Advant® OCS
with Master software

Engineering Methods

User’s Guide

3BSE 013 933R201

Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

Advant, AdvaCommand, AdvaInform, and AdvaBuild are registered trademarks of ABB Asea Brown Boveri Ltd., Switzerland.

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Automation Products GmbH. ABB Automation Products GmbH assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Automation Products GmbH be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Automation Products GmbH be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Automation Products GmbH’s written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

Copyright © ABB Automation Products GmbH 1999

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 Before You Begin.......... 1-1
1.2 Equipment Requirements.......... 1-1
1.3 How to Use this Book.......... 1-1
1.4 Related Documentation.......... 1-2
1.5 Release History.......... 1-3

Chapter 2 - Structuring of Engineering Information

2.1 Advant Objects.......... 2-1
2.2 Aspects of Advant Objects.......... 2-3
2.3 Views on Aspect Data of Advant Objects.......... 2-4
2.4 Relations Between Advant Objects 2-5
  2.4.1 Functional Relations Between Advant Objects 2-8
  2.4.2 Locational Relations Between Advant Objects 2-8
  2.4.3 Relations Between Advant Objects of the Control System 2-10

Chapter 3 - Example Project and Example Workflow

3.1 Overview of the Example Project 3-1
3.2 Basic Information to Start with.......... 3-1
3.3 Example Workflow 3-2

Step 1 - Build Project Environment

S1.1 Creating a Project.......... S1-1
S1.2 Assigning of Users to the Project S1-2
S1.3 Opening the Project and Assigning the Project Global Data.......... S1-3

Step 2 - Build Library Environment

S2.1 Installation of the Object Library Project S2-1
  S2.1.1 Setup the File System from the CD S2-1
  S2.1.2 Import the Object Library Project.......... S2-2
S2.2 Structure and Content of the Object Library S2-3
  S2.2.1 Function Oriented Structure.......... S2-3
  S2.2.2 Control System Oriented Structure.......... S2-5
  S2.2.3 Mounting Location Structure.......... S2-5
S2.3 Assigning Articles to the Object Library.......... S2-6
  S2.3.1 Creating the Articles in the Object Library S2-8

CONTENTS (continued)

Step 3 - Build the Basic Function Oriented Structure

S3.1 Opening the Function Oriented Structure S3-1
S3.2 Placing the Project Object in the Function Oriented Structure S3-2
S3.3 Creation of Advant Objects in the Function Oriented Structure S3-2
S3.4 Assigning Engineering Data to Advant Objects in the Basic Structure S3-5
  S3.4.1 Opening an Advant Object S3-5
  S3.4.2 Creation of Aspects S3-6
    S3.4.2.1 Automatically Created Aspects S3-6
    S3.4.2.2 User Added Aspects S3-8
    S3.4.2.3 Editing Aspect Data S3-9
    S3.4.2.4 Aspect Description and Administration S3-13

Step 4 - Build the Basic OCS-Structure

S4.1 Opening the Control System Structure S4-1
S4.2 Placing the Project Object in the Control System Structure S4-2
S4.3 Creating Objects in the Basic Control System Structure S4-2
S4.4 Setting the OCS Defaults S4-3

Step 5 - Build the Location Structure

5.1 Opening the Location Structure S5-1
S5.2 Placing the Project Object in the Location Structure S5-2
S5.3 Creation of Objects in the Location Structure S5-3
S5.4 Assigning Engineering Data to Location Objects S5-4

Step 6 - Build New / Adapt Standard Solutions

S6.1 Working with Standard Solutions S6-1
S6.2 Creating a Standard Control Loop S6-2
  S6.2.1 Advant Objects of the Standard Control Loop in the Object Library ... S6-2
  S6.2.2 Examples of Advant Object Aspects in the Standard Control Loop S6-3
    S6.2.2.1 Document Aspect S6-3
    S6.2.2.2 Circuit Aspect S6-4
    S6.2.2.3 Electrical Circuit Diagram Aspect S6-5
S6.3 Creating a Standard Drive System S6-6
  S6.3.1 Advant Objects of the Standard Drive System in the Object Library ... S6-6
  S6.3.2 Examples of Advant Object Aspects in the Standard Drive System ... S6-12
    S6.3.2.1 Aspects Related to the Electrical Diagram Builder S6-12
    S6.3.2.2 OCS Related Aspects S6-15
  S6.3.3 Assigning Articles- (Product-) Data to Advant Object Aspects S6-15

CONTENTS (continued)

S6.4 Importing Standard Solutions into a Library S6-19
S6.5 Using Standard Solutions........ S6-21
  S6.5.1 Copying and Importing Standard Solutions to Application Projects.. S6-21
  S6.5.2 Placing the Objects of a Solution in the Application Project S6-22

Step 7 - Build Process Object Data

S7.1 Creating the Process Object List........ S7-1

Step 8 - Build Signal Data

S8.1 Creating the Signal List S8-1

Step 9 - Import of Process Object Data

S9.1 Import Procedure S9-1
S9.2 Import Results........ S9-2
  S9.2.1 Creation of Process Objects in the Target Project S9-2
  S9.2.2 Aspects and Parameters of Imported Process Objects........ S9-3
  S9.2.3 Reimport of Process Object Data........ S9-4

Step 10 - Import of Signal Data

S10.1 Import Procedure S10-1
S10.2 Import Results........ S10-2
  S10.2.1 Creation of Signals in the Target Project S10-2
  S10.2.2 Aspects and Parameters of Signal Objects S10-3
  S10.2.3 Reimport of Signal Data S10-4

Step 11 - Build PC-Program

S11.1 Create Node Control Software Aspect........ S11-1
S11.2 Create PC-Program in Node Control Software Aspect S11-1

Step 12 - Perform I/O Allocation

S12.1 Drag and Drop Signals to Channels........ S12-1
S12.2 Placing Signals of the Drive System Example S12-1

Step 13 - Build/Adapt Modules

S13.1 Create Circuit Aspect........ S13-1
S13.2 Create Control Module in Circuit Aspect........ S13-1

Step 14 - Instantiate SW Modules

S14.1 Using Circuits in the Node Mode S14-1

CONTENTS (continued)

Step 15 - Interconnect SW Modules

S15.1 Create Glue Logic S15-1

Step 16 - Perform Parameter Update

S16.1 Summary of Parameter Update Procedures S16-1
S16.2 Updating Parameters in the Function Chart of a Circuit S16-1
S16.3 Updating the Parameters for a Complete Project S16-2

Step 17 - Generate DB Elements

S17.1 Generate DB Elements for Node Control Software S17-1

Step 18 - Build Mounting Locations

S18.1 Engineering Related to Mounting Locations S18-1
    S18.1.1 Building the Mounting Location Structure S18-1
    S18.1.2 Working with Standard Solutions of Mounting Locations S18-2
    S18.1.3 Building a Mounting Location Structure from Diagrams S18-6

Step 19 - Arrange Components and Devices

S19.1 Placing of Devices and Components in Locations S19-1
S19.2 Placing Objects of the Drive System Example S19-1

Step 20 - Build/Adapt Circuit Diagrams

S20.1 Create Electrical Circuit Diagram S20-1
S20.2 Edit Electrical Circuit Diagram S20-2

Step 21 - Update Circuit Diagrams

S21.1 Preconditions S21-1
S21.2 Synchronization Structure Builder-Electrical Diagram Builder S21-1
    S21.2.1 Save Board/Connection Unit Circuit Diagrams to Database S21-2
    S21.2.2 Save I/O Circuit Diagrams to Database S21-3
    S21.2.3 Save Circuit Diagrams for Consumers to Database S21-4
    S21.2.4 Synchronize CAEE S21-5
    S21.2.5 Update Circuit Diagrams Board-Connection Unit from Database S21-5
    S21.2.6 Update I/O Circuit Diagrams from Database S21-6
    S21.2.7 Update Circuit Diagrams for Consumers from Database S21-6

Appendix A - Designation of Advant Objects

A.1 Applying the Reference Designation A-1
A.2 Presentation of the Absolute Reference Designation A-2

ILLUSTRATIONS

Figure 2-1. Examples of Advant Objects Used in Electrical and Control Engineering.. 2-2
Figure 2-2. Illustration of an Advant Object with Its Aspects 2-3
Figure 2-3. Advant Objects and Its Aspects.. 2-4
Figure 2-4. Illustration of Advant Object, Aspects, Views 2-5
Figure 2-5. Hierarchical Relations Between Advant Objects 2-6
Figure 2-6. Relations Between Advant Objects 2-7
Figure 2-7. Principle of a Function Oriented Structure.. 2-8
Figure 2-8. Principle of a Location Oriented Structure.. 2-9
Figure 2-9. Principle of a Control System Oriented Structure...................................... 2-10
Figure 3-1. Example Workflow.. 3-3
Figure S1-1. Project Administration Window S1-2
Figure S1-2. Assign Users to Project Window... S1-3
Figure S1-3. Editing Project Global Data Aspect S1-4
Figure S2-1. Project Administration Window Showing Exported Projects S2-3
Figure S2-2. Basic Structure of Function Items S2-4
Figure S2-3. Example of a Control Loop and a Drive System in the Object Library S2-4
Figure S2-4. Control System Structure in the Object Library.. S2-5
Figure S2-5. Basic Mounting Location Structure of the Object Library........................ S2-6
Figure S2-6. View on Contents of Articles Library.. S2-6
Figure S3-1. Opening the Function Structure in the Open Structure Window............... S3-1
Figure S3-2. Function Structure Window with Placed Project Object........................... S3-2
Figure S3-3. Creation of a New Object S3-4
Figure S3-4. Higher Levels of the Function Structure S3-4
Figure S3-5. Project Pool S3-5
Figure S3-6. Project Object Window.. S3-6
Figure S3-7. Editing Attributes of Basic Properties... S3-7
Figure S3-8. Adding an Aspect to an Advant Object... S3-8
Figure S3-9. Opening a View on a Document Aspect.. S3-10
Figure S3-10. View on the Summary Form of a Document Aspect............................... S3-11
Figure S3-11. Document Aspect of the Advant Object "Spray Cooling System" S3-12
Figure S3-12. Document View Presented in MS Word.. S3-13
Figure S3-13. Opening the Aspect Registry Box S3-14
Figure S3-14. Structure Window with Object Area and Aspect Area............................ S3-15
Figure S4-1. Opening the OCS-Structure in the Open Structure Window..................... S4-1
Figure S4-2. Basic Structure of a Control System S4-3
Figure S4-3. Setting the OCS Defaults.. S4-4
Figure S5-1. Opening the Location Structure in the Open Structure Window................ S5-1
Figure S5-2. Location Structure Window with Placed Project Object........................... S5-2

ILLUSTRATIONS (continued)

Figure S5-3. Location Structure for the Example Project S5-4
Figure S5-4. Overview of Arrangement and Interconnection S5-5
Figure S5-5. Open Object Window with Aspect for Arrangement and Interconnection. S5-6
Figure S6-1. Standard Flow Control Loop in the Object Library.................................. S6-2
Figure S6-2. Selecting a Document View for Editing S6-3
Figure S6-3. Example for the Document View of a Document Aspect.......................... S6-3
Figure S6-4. Adding a Circuit Aspect to an Advant Object S6-5
Figure S6-5. Standard Drive System in the Object Library... S6-6
Figure S6-6. Object DS.P1 with Its Aspects.. S6-7
Figure S6-7. Object DS.C1_MCC Binder with Its Aspects S6-8
Figure S6-8. Object DS.P1_IO Binder with Its Aspects.. S6-9
Figure S6-9. Object DS.P1_Main_Drive Motor with Its Aspects S6-10
Figure S6-10. Object DS.C1_RUN_R I/O-Signal with Its Aspects S6-11
Figure S6-11. Form View of Electrical Structure Aspect Data S6-12
Figure S6-12. Form View of Electrical Device Aspects... S6-13
Figure S6-13. Datasheet View of Electrical Function Aspects...................................... S6-13
Figure S6-14. Typicals of Electrical Circuit Diagrams.. S6-14
Figure S6-15. Signal Parameters S6-15
Figure S6-16. Electrical Articles Assigned to El. Device Aspects S6-16
Figure S6-17. Referencing Article Data S6-18
Figure S6-18. Importing a Solution S6-20
Figure S6-19. Example for the Solution of a Cooling System S6-22
Figure S7-1. Process Object List S7-2
Figure S7-2. Comma Delimited Format of a Process Object List S7-2
Figure S7-3. Process Object in the Object Library... S7-3
Figure S8-1. Signal List... S8-1
Figure S8-2. Comma Delimited Format of a Signal List.. S8-2
Figure S8-3. Signal Object in the Object Library Placed in two Structures.................... S8-2
Figure S9-1. Import Dialog.. S9-1
Figure S9-2. Import of Process Objects... S9-2
Figure S9-3. Imported Process Object with View on its Parameter Aspect S9-3
Figure S10-1. Import Dialog.. S10-1
Figure S10-2. Import of Signal Objects... S10-2
Figure S10-3. Imported Signal Objects S10-3
Figure S10-4. View on the Attributes of a Signal Parameter Aspect.............................. S10-3
Figure S11-1. Inserting a PC-Program in a Node Control Software Aspect S11-2
Figure S12-1. Placing Signals.. S12-1
Figure S13-1. Inserting a Control Module into a Circuit Aspect................................... S13-2

ILLUSTRATIONS (continued)

Figure S14-1. Instantiating the Circuit Aspect of an Object by Drag and Drop S14-2
Figure S15-1. Editing a Node Function Chart.. S15-2
Figure S16-1. Updating the Parameters of a Circuit S16-2
Figure S16-2. Updating the Parameters for a Complete Project S16-3
Figure S17-1. Generation of all DB Elements of a Node... S17-2
Figure S18-1. Higher Levels of a Mounting Location Structure.................................... S18-2
Figure S18-2. Structure of a Standard Control System Cabinet...................................... S18-3
Figure S18-3. Copying a Standard OCS-Cabinet Assembly.. S18-4
Figure S18-4. Object Window of an OCS-Cabinet S18-5
Figure S18-5. Examples for Diagram Aspects Assigned to an Object............................ S18-7
Figure S18-6. Saving a Diagram to the Database of the El. Diagram Builder................ S18-8
Figure S18-7. Synchronizing El. Diagram Builder to Structure Builder Database......... S18-9
Figure S19-1. Placing the OCS-Boards in the Mounting Location Structure S19-1
Figure S19-2. Placing of Binders in the Mounting Location Structure........................... S19-2
Figure S19-3. Placing the Motorcontroller in a Cabinet S19-3
Figure S19-4. Placing the Motor in the Topographic Location Structure S19-3
Figure S20-1. Binder Objects in the Function Structure S20-1
Figure S20-2. Editing an Electrical Circuit Diagram S20-2
Figure S21-1. Synchronizing Structure Builder to El. Diagram Builder Database......... S21-2
Figure S21-2. Circuit Diagram Board-Connection Unit S21-3
Figure S21-3. I/O Circuit Diagram Showing Two Digital Input Channels S21-4
Figure S21-4. Circuit Diagram of a Drive System S21-5
Figure S21-5. Information Related to Channels in I/O Circuit Diagrams....................... S21-6
Figure A-1. Editing the Relative Reference Designation...A-2
Figure A-2. Relative and Absolute Reference Designation A-3
Figure A-3. Display Variants for the Absolute Reference Designation A-4

TABLES

Table 1-1. Related Documentation.......... 1-2
Table 1-2. Versions.......... 1-3
Table S6-1. Typical Placements of Solution Objects in an Application S6-23

Chapter 1 Introduction

1.1 Before You Begin

This manual is intended as a guide for plant engineering using AdvaBuild Engineering Software. Instructions for operating the various Builders are provided in other manuals, described in Section 1.4, Related Documentation.

1.2 Equipment Requirements

For a description of the system requirements consult the *Advant Workplace* and *AMPL Control Configurations Release Notes* and the Release Notes of the individual Builders.

1.3 How to Use this Book

This book is organized into the following chapters:

- Chapter 1 Introduction
- Chapter 2 Structuring of Engineering Information
- Chapter 3 Example Project and Example Workflow.

The parts of the book that follow these chapters are structured into 21 working steps (Step 1 to Step 21) and an appendix. The working steps are defined in Chapter 3.

1.4 Related Documentation

Table 1-1 lists documentation related to the content of this reference manual.

Table 1-1. Related Documentation

Title	Description
Structure Builder User’s Guide	Describes the creation of projects, Advant objects, structures and aspects, and how to navigate in projects.
Control Configuration Functions User’s Guide	Describes the integration of the Application Builder, Function Chart Builder and On-line Builder into the Structure Builder.
Application Builder User’s Guide	Describes the installation of functions and components for control configuration and contains the operating instructions of Application Builder, Bus Configuration Builder and support functions.
Function Chart Builder User’s Guide	Contains a description of AMPL application programs configuration.
Parameter Builder User’s Guide	Describes how to handle parameters for process objects and for OCS objects and how to import engineering data.
Document Builder User’s Guide	Describes the support of handling various types of documents, including the handling of templates.
Electrical Diagram Builder User’s Guide	Describes the electrical plant engineering and the handling of circuit diagrams
Wiring Builder User’s Guide	Describes how to create and edit electrical wiring data of wires and cables assigned to physical connections between devices.
Object Library Reference Manual	Describes the content of the Object Library that is delivered with the AdvaBuild Engineering Software.

1.5 Release History

This is the second version of the Engineering Methods User's Manuals.

Table 1-2. Versions

Version	Description
3BSE013933R0001	Describes the principles of engineering based on the Advant object concept.
3BSE013933R0101	Describes the principles of engineering based on the Advant object concept, including electrical plant engineering.
3BSE013933R0201	Describes the principles of engineering based on the Advant object concept, including electrical plant engineering adapted to CAEE version 3.1/1.

Chapter 2 Structuring of Engineering Information

2.1 Advant Objects

All information handled during the engineering process is assigned to **Advant objects**.

An Advant object is a data model for an entity treated in a design, engineering, operation and maintenance process. Advant objects are, for example

- Main plant equipment like:
  - Casting machine
  - Rolling mill stand
- Components and devices like:
  - Motors
  - Cabinets
  - I/O-boards
- Agglomerations of Advant objects that fulfill a certain function like:
  - Drive system
  - Control loop
- Information entities like:
  - Signals
- Location entities like:
  - Areas, Buildings and other locations to which plant equipment can be assigned.

The Advant objects handled in the engineering process represent physical objects or virtual objects, for example, functions, and are used as a container for engineering data. **AdvaBuild Engineering Software supports a concept of plant modeling by using Advant objects.**

Figure 2-1 shows typical examples of Advant objects used in electrical and control engineering.

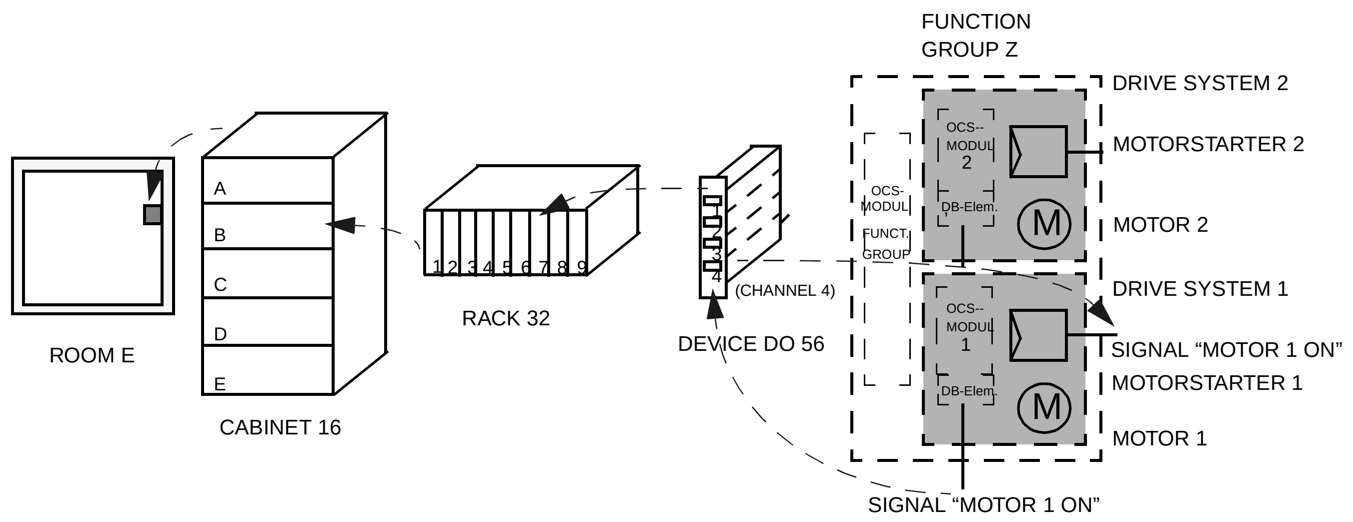


Figure 2-1. Examples of Advant Objects Used in Electrical and Control Engineering

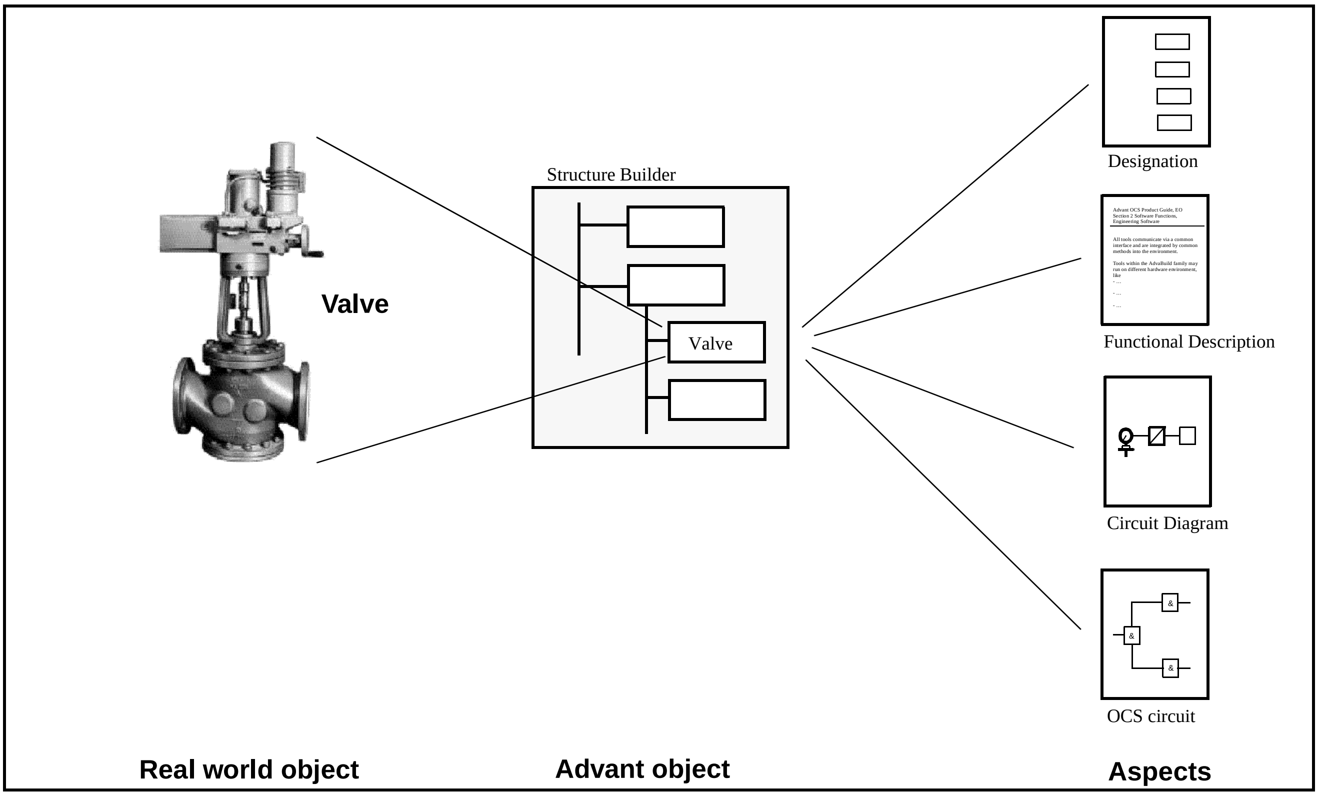
2.2 Aspects of Advant Objects

The engineering data describing an Advant object is grouped into aspects. An aspect is an assembly of data that describe certain properties of an Advant object, such as:

- Function properties
- Physical construction properties
- Location properties.

An Advant object may have several aspects, it has at least one aspect which describes the Basic Properties of the Advant object, for example, name and type. Each aspect is handled by assigned tools.

Figure 2-2 illustrates the definitions Advant object and aspect.



Advant OCS Product Guide, EO
Section 2 Software Functions,
Engineering Software

All tools communicate via a common interface and are integrated by common methods into the environment.

Tools within the AdvaBuild family may run on different hardware environment, like

- ...
- ...
- ...

Figure 2-2. Illustration of an Advant Object with Its Aspects

Figure 2-3 shows the examples of Advant objects of Figure 2-1 supplemented by its aspects.

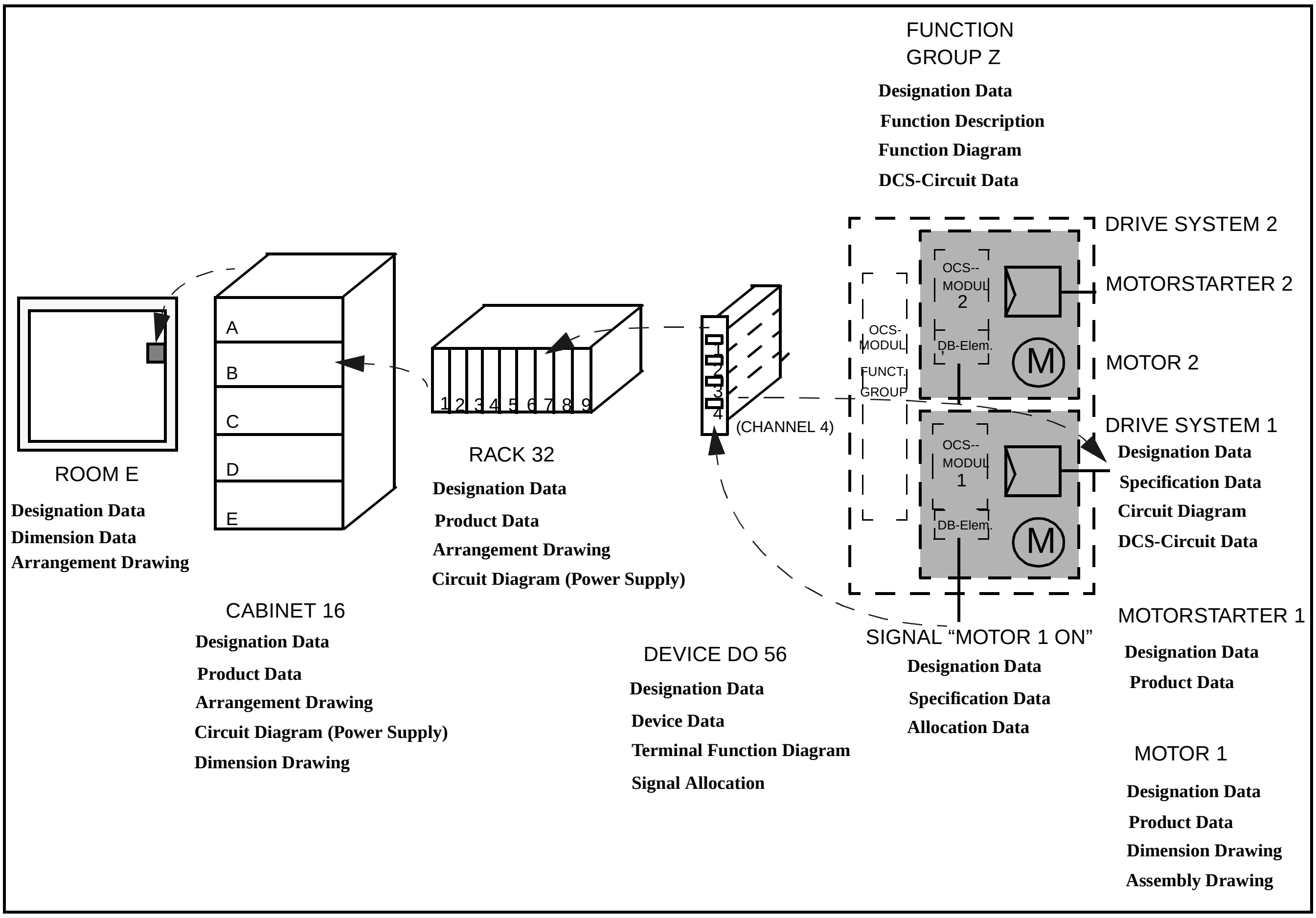


Figure 2-3. Advant Objects and Its Aspects

2.3 Views on Aspect Data of Advant Objects

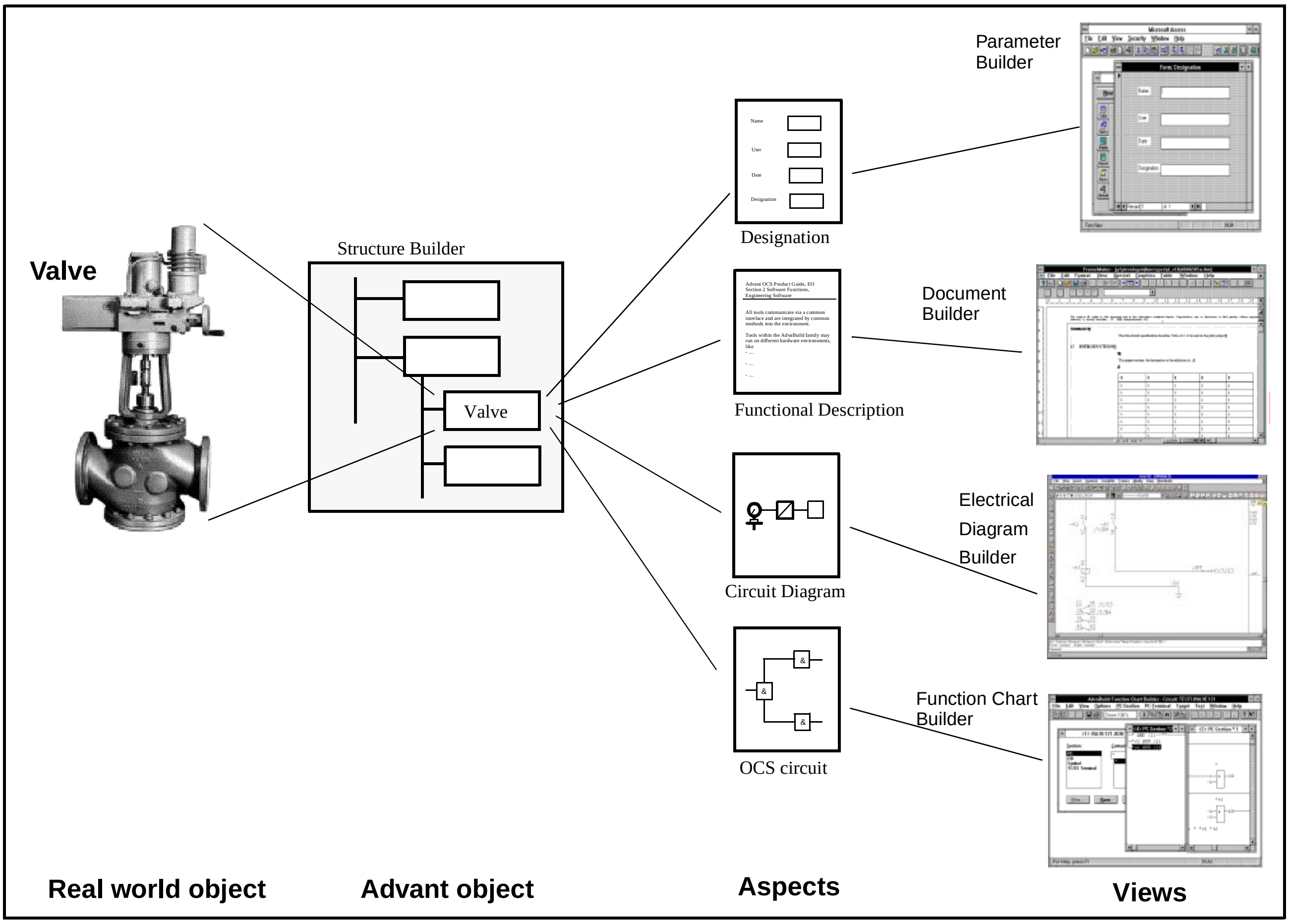
Aspect data are presented to the user in views. A view is a specific presentation of engineering and administration data belonging to one or several Advant objects. The presentation in a view gives the possibility to inspect, modify and map engineering data.

The data and their mapping may be presented, for example, as:

- List
- Table
- Diagram (Circuit Diagram)
- Drawing (Arrangement Drawing)

Views are predefined in AdvaBuild Engineering Software but may also be defined by the user.

Views are assigned to aspects of Advant objects. An aspect may have assigned several views. Opening aspect views by the Structure Builder gives access to the relevant aspect system with the relevant tool for editing the data.



Advant OCS Product Guide, EO
Section 2 Software Functions,
Engineering Software

All tools communicate via a common interface and are integrated by common methods into the environment.

Tools within the AdvaBuild family may run on different hardware environments, like
- ...
- ...
- ...

Figure 2-4. Illustration of Advant Object, Aspects, Views

2.4 Relations Between Advant Objects

Advant objects may be integrated into hierarchical relations with regard to their:

- Function properties
- Location properties
- Product properties
- Control system properties.

The following example shows an Advant object representing a drive system with its constituent Advant objects:

- Motor
- Motorstarter
- Power Cable.

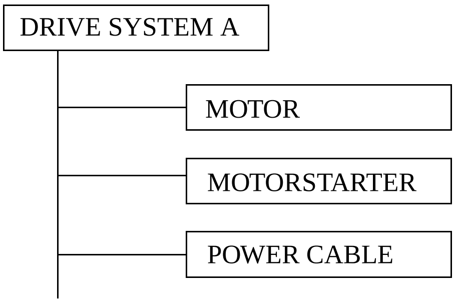


Figure 2-5. Hierarchical Relations Between Advant Objects

Further relations between Advant objects are of the type

- "is located in" (a product in a location as a board in a rack slot)
- "is connected to" (a signal to a channel, a wire to a terminal)
- "runs in runtime unit" (a control algorithm for drive system in a controller node)

Advant objects can be created in certain structures and are always placed additionally into a pool. Objects that are only placed in the object pool are unrelated to each other.

The relations between Advant objects are established by placing them in one of the following predefined structures for Advant objects:

- Function Structure
- Location Structure
- Mounting Location Structure
- Control System Structure
- Product Structure.

Placing of Advant objects into these predefined structures supports the navigation in the engineering data and the designation of the Advant objects in the plant and the documents.

You may also create and use your own structures.

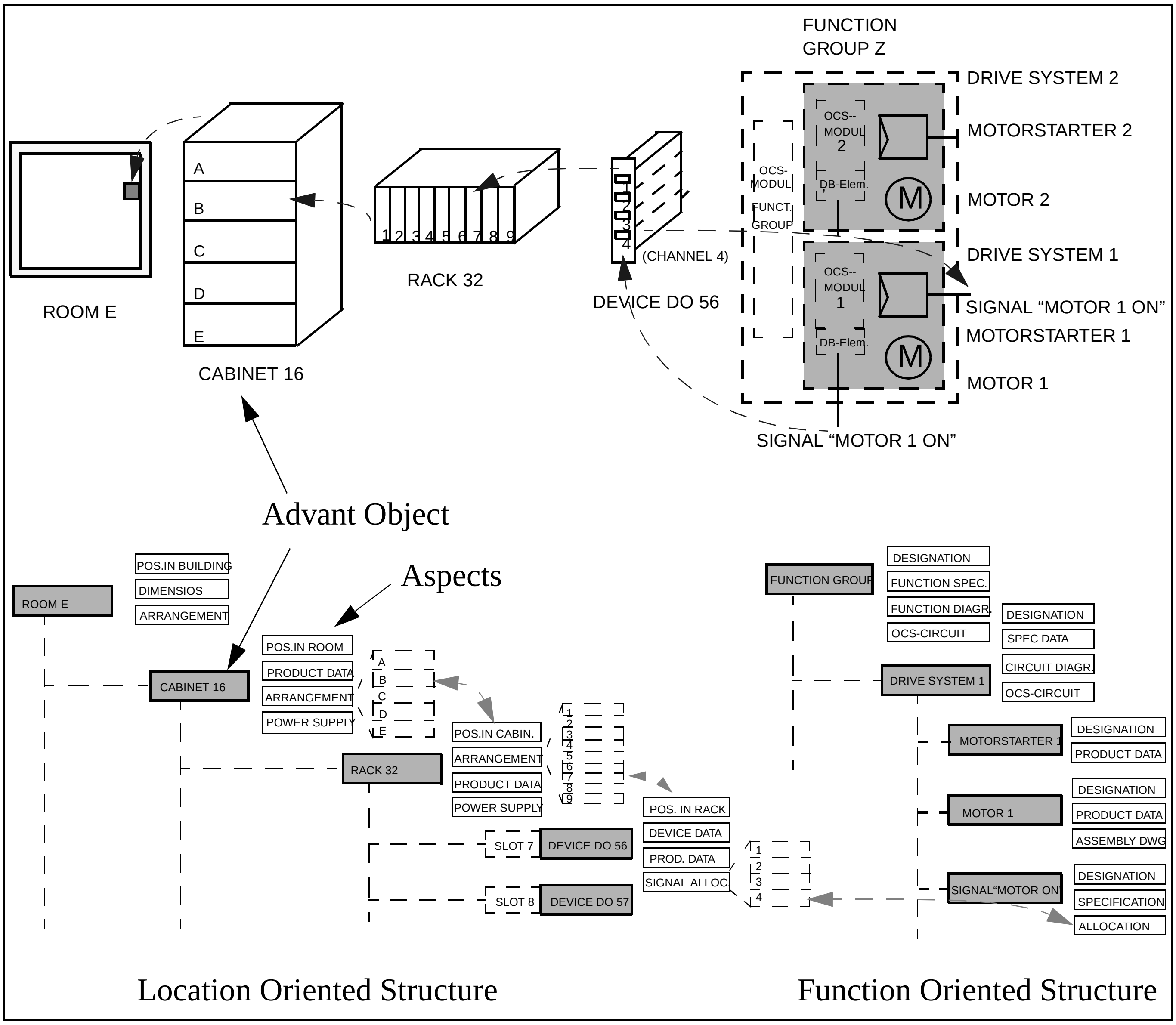


Figure 2-6. Relations Between Advant Objects

2.4.1 Functional Relations Between Advant Objects

Functional relations between Advant objects are presented in a function oriented structure. This structure is build up by placing Advant objects according to their hierarchical relations into this structure.

Figure 2-7 shows the principles of a function oriented structure.

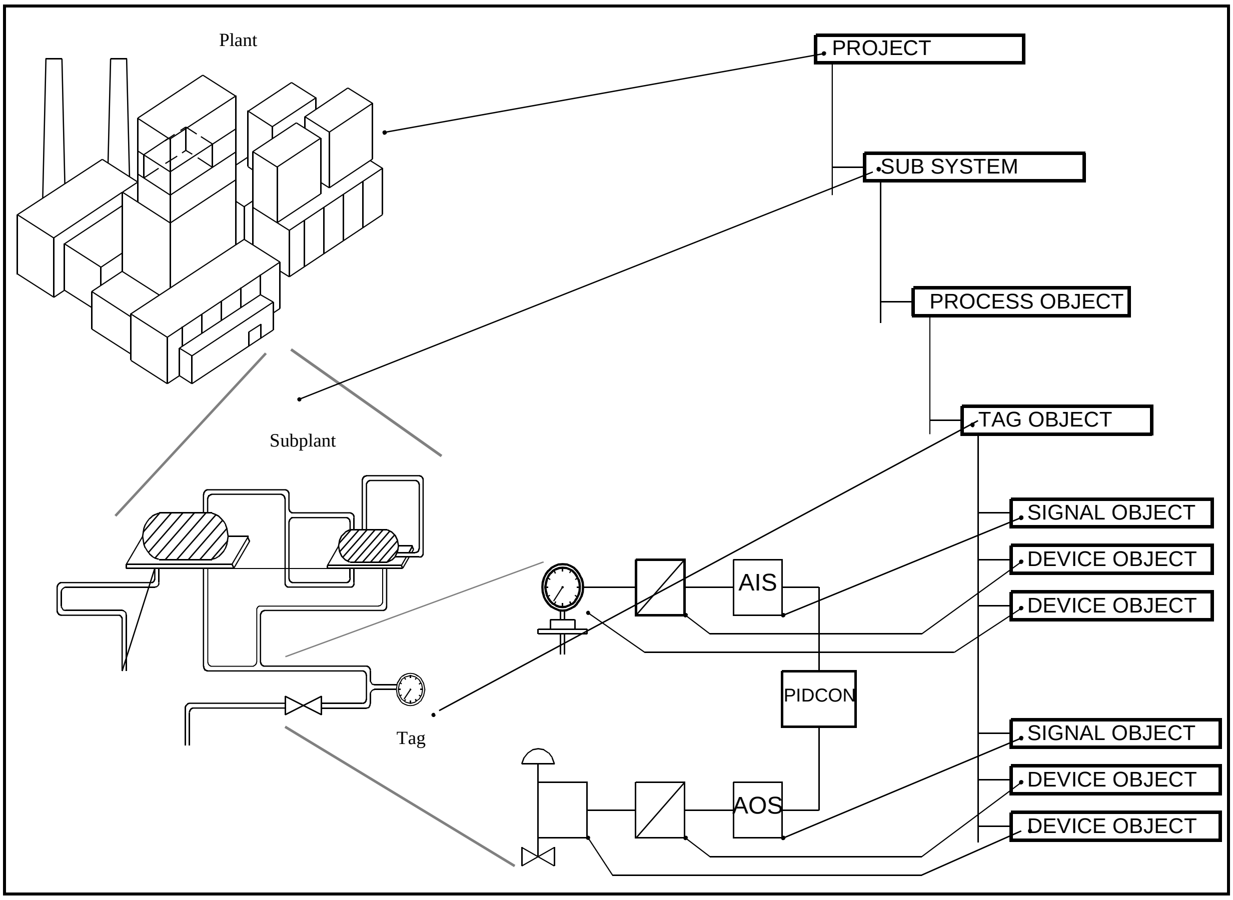


Figure 2-7. Principle of a Function Oriented Structure

2.4.2 Locational Relations Between Advant Objects

Locational relations between Advant objects are presented in a location oriented structure. Distinguish between topographic locations like:

Areas, Buildings, Rooms

And mounting locations like:

Cabinets, Desks, Racks, Slots.

The topographic location structure and the mounting location structure can be created independently from each other and connected in a later planning stage. The connection is performed by placing mounting location structure trees into the topographic location structure. The mounting location structure contains a number of Advant object structure trees that represent for example the cabinets, desks and other constructional units of the plant.

Figure 2-8 shows the principles of a location oriented structure.

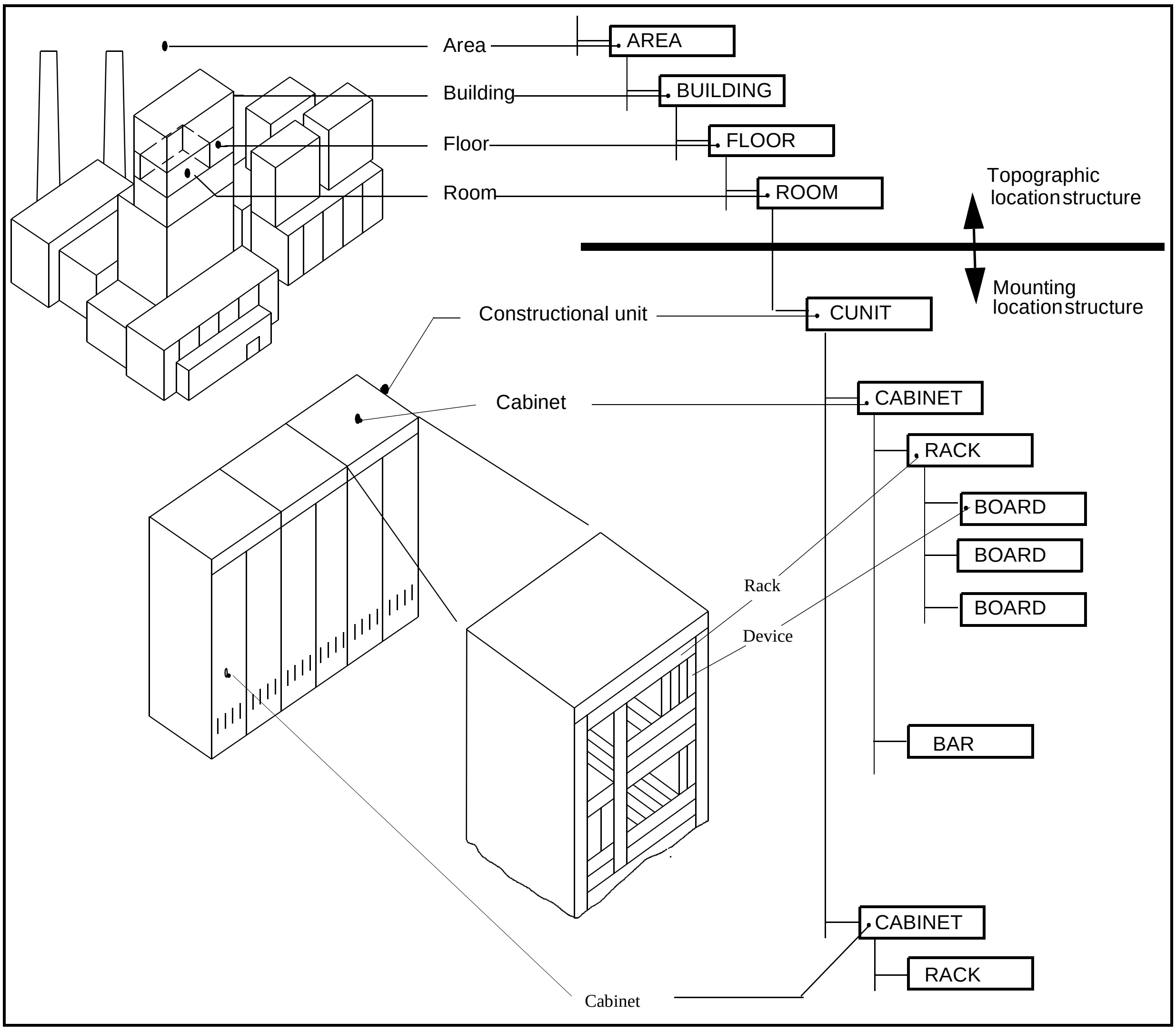


Figure 2-8. Principle of a Location Oriented Structure

2.4.3 Relations Between Advant Objects of the Control System

The relations between Advant objects representing constituents of the control system are presented in a control system structure. The structure is built up by placing the relevant Advant objects into this structure type.

Figure 2-9 shows the principles of structuring Advant objects that are constituents of the control system.

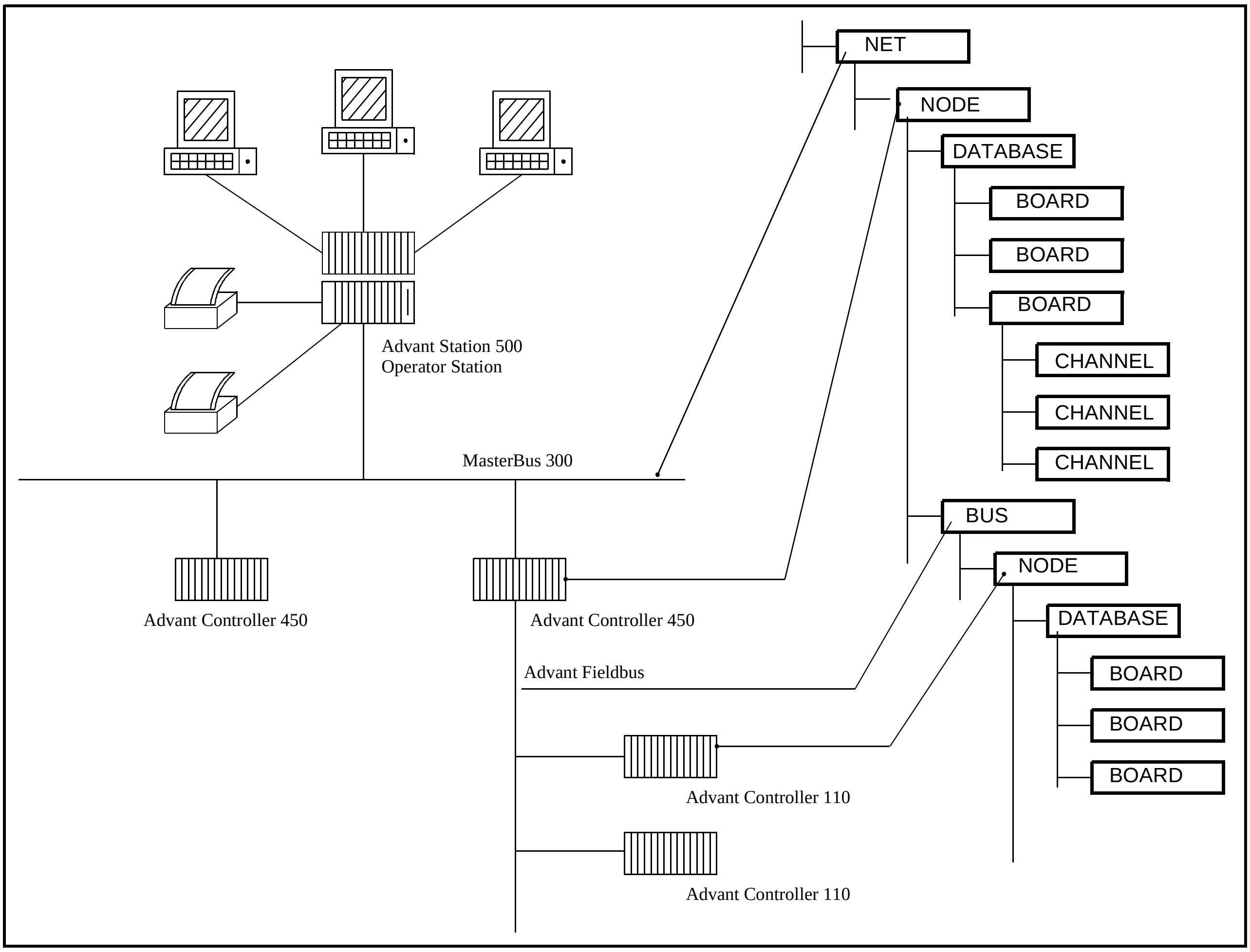


Figure 2-9. Principle of a Control System Oriented Structure

Chapter 3 Example Project and Example Workflow

3.1 Overview of the Example Project

A cooling system within a continuous casting plant is used as an example to explain the engineering process in this manual. The result of this engineering process shall be all the data and documents that are needed for:

- Installation,
- Commissioning
- Operation
- Maintenance.

There are several cooling systems in a continuous casting plant:

- Mould cooling system
- Machine cooling systems
- Strand spray cooling system.

The engineering task considered here is the engineering of the strand spray cooling system.

This cooling system cools the casted strand that got its shape in the mould but is still fluid below its surface shell. The cooling system has the purpose to cool the strand down to a completely solid state.

The spray cooling system is divided into four cooling zones. Each zone contains a flow control loop for cooling water and a pressure monitoring system for the cooling water pressure. The cooling water is sprayed via several nozzles per cooling zone onto the surface of the casted strand. The controlled water flow per zone is split up to these nozzles

The set point for the cooling water flow in each zone is dependent from the casting speed. The relations cooling water versus casting speed is documented in tables.

In our example there are four individual controlled spray cooling zones. The water pressure for all cooling zones is generated by one pump which is driven by an electrical drive system.

A further example is demonstrated on a conveyor system that transports the steel strands.

3.2 Basic Information to Start with

The scope of supply and the necessary engineering task for a project is usually defined in a contract. The basic input data on which the engineering is started are available in documents from the customer or mechanical partner. These documents describe the plant, the processes running the plant and specify basically the functions that shall be implemented to run the processes.

Plant and process describing documents are, for example:

- Plant Layout
- Process Diagram and Process Description.

Documents that specify the basic functions are, for example:

- P&I-Diagram
- List of Measuring Points
- Motor and Component List.

These documents are delivered by the customer or partner with a certain status and may get further revisions during the engineering process. These revisions of the documents that specify the basic functions are normally handled by the supplier of the electrical and control equipment in cooperation with the customer.

3.3 Example Workflow

The example project will be executed according to an example workflow as shown in Figure 3-1. This example workflow covers mainly the following activities:

- Complete Function Description
- Perform standard design
- Perform new and adapted design
- Final design
- Software production.

The example workflow shows activities and their input/output relations. Each activity can be assigned to one of the following engineering branches:

- Function oriented engineering
- Location oriented engineering
- Control system oriented engineering.

These engineering branches are reflected in corresponding structure types of the plant engineering model.

This workflow includes also the creation of an Advant object library for a project. This is an extract of a general reference library and contains only those solutions and components that shall be used for the relevant project to be executed. New created or adapted solutions will be put into these library besides the components and solutions already available in the start-up status of the library. In our example we start with the import of a basic Object library to be used for the example project.

The import of data that were created externally is managed by import functions.

In our example project we will refer to the activities and to the main branches of the example workflow. The focus is put on the workflow for the function oriented and control system oriented engineering. Figure 3-1 shows the example workflow. Each step in the workflow of Figure 3-1 is described in a corresponding section of this manual.

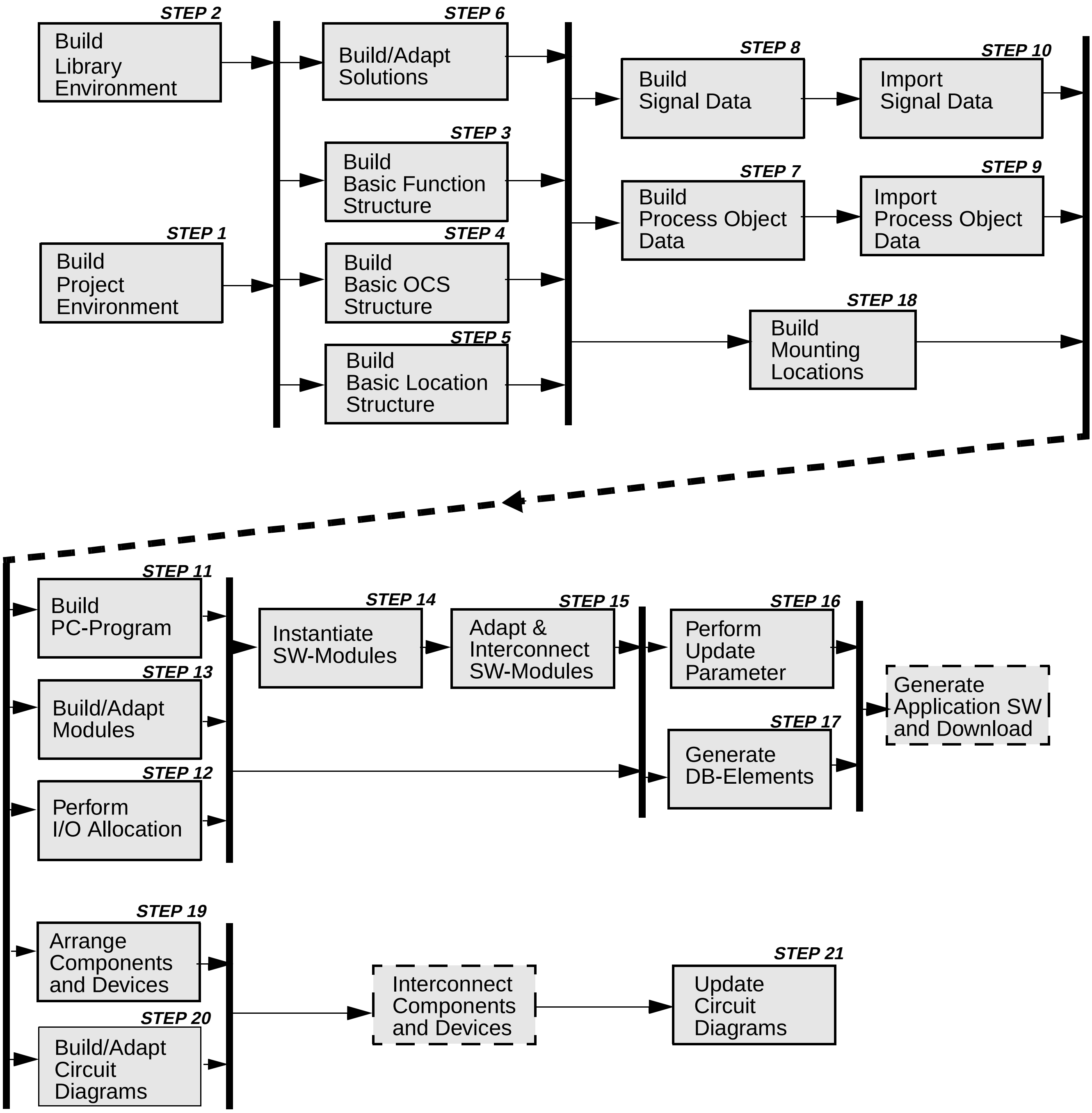


Figure 3-1. Example Workflow

Step 1 Build Project Environment

The project environment is built on a system that has all the builders, required for the execution of the project, installed. The main steps in building the project environment are:

- Creating a project on an Advant Engineering Workplace
- Assigning of users to the project
- Assigning of the project global data.

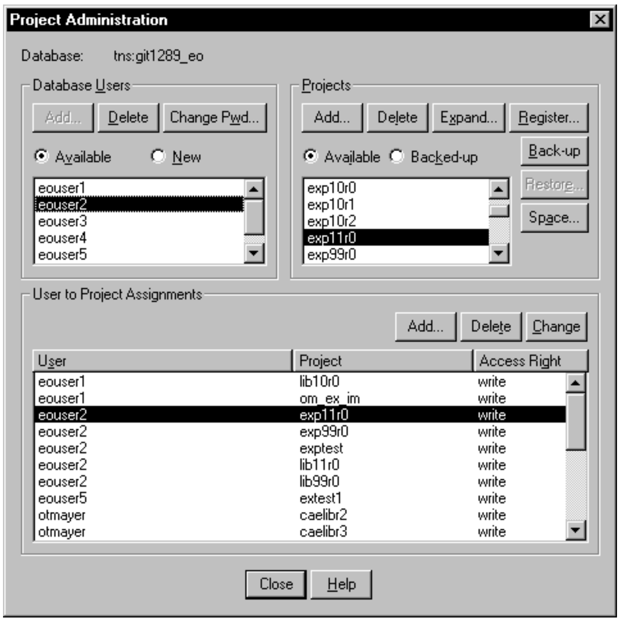
S1.1 Creating a Project

To create a project, log into the system and open the Administration component of the Structure Builder.

Creating of a Project	
Preconditions:	Access rights available
User activities:	1. Start up the Structure Builder 2. Choose **Administration** from the **Project** menu 3. Enter the connect string **tns:*servername*_eo** for the database and enter your administrator password in the **Open Database** box and click on **OK** 4. Press **Add** in the **Projects** group box in the **Projects Administration** window 5. Insert Project name in the **Add Project** window 6. Edit database size for CAEE and STB 7. Click on **OK**
Result:	New project is created

When a project is created, an object that represents this project is created. This object is put into the object pool and will later be used as top object in the main project structures.

Figure S1-1 shows the Project Administration window.



Database: tns:git1289_eo

eouser1
eouser2
eouser3
eouser4
eouser5

exp10r0
exp10r1
exp10r2
exp11r0
exp99r0

User	Project	Access Right
eouser1	lib10r0	write
eouser1	om_ex_im	write
eouser2	exp11r0	write
eouser2	exp99r0	write
eouser2	exptest	write
eouser2	lib11r0	write
eouser2	lib99r0	write
eouser5	extest1	write
otmayer	caelibr2	write
otmayer	caelibr3	write

Figure S1-1. Project Administration Window

S1.2 Assigning of Users to the Project

Before users can be assigned to a project they must have been assigned to the Windows NT group AdvaBuild Engineers. These users are shown as available users in the Users group box of the Project Administration window, see Figure S1-1. If you click on the radio button New in the Users group box you see the members of the AdvaBuild Engineering group that are not yet mapped to the database just opened

Assigning Users to a Project	
Preconditions:	Project Administration window open
User activities:	1. Press **Add** in the **Users to Project** Assignments group box in the **Projects Administration** window 2. Assign User to Project in the **Assign Users to Project** window and select a button in **Access Rights** 3. Click on **OK** 4. Repeat steps 1 to 4 for additional users
Result:	User assigned to Project

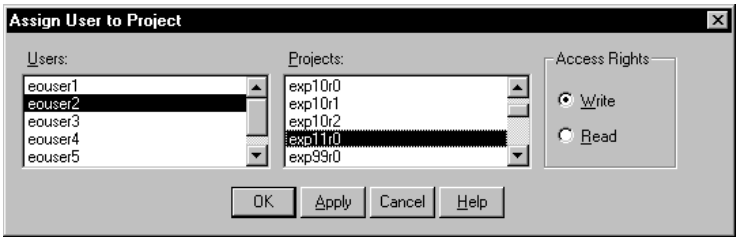


Figure S1-2. Assign Users to Project Window

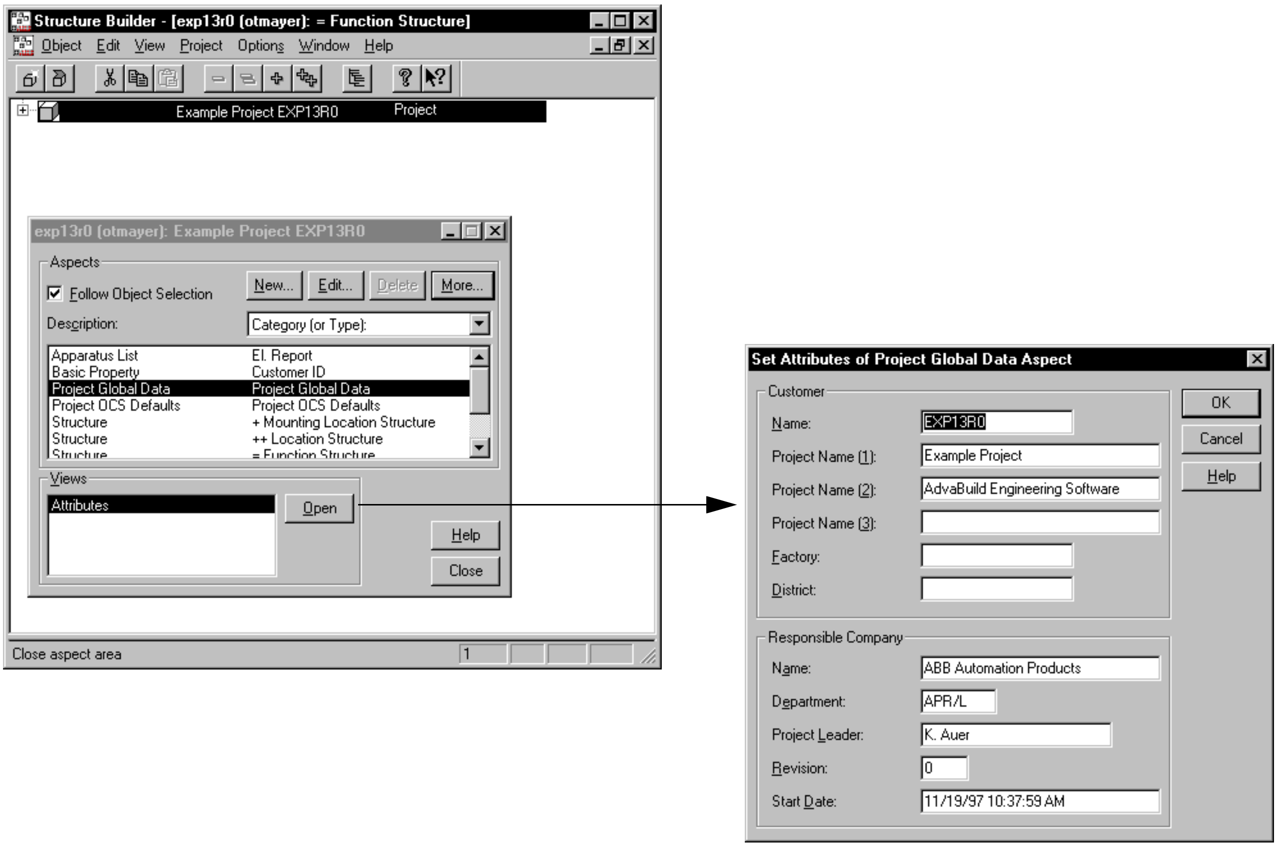
S1.3 Opening the Project and Assigning the Project Global Data

Open the project pool window to get access to the new created project object.

Opening a New Created Project	
Preconditions:	User assigned to project
User activities:	1. Use the Structure Builder 2. Choose **Open** from the **Project** menu 3. Insert database connect string **tns:*servername*_eo**, project name, user name and password 4. Click on **OK** 5. Select **Project Pool** in the **Open Structure** window 6. Click on **OK** 7. Click on the **Project** object in the **Project Pool** window 8. Choose **Open** from the **Object** menu 9. Choose **Project Global Data** aspect in the **Object** window 10. Open the **Attributes** view 11. Edit the Project Global Data in the **Set Attributes of Project Global Data Aspect** window 12. Click on **OK**
Result:	Project is open and Project Global Data attributes set

Because it is a new created project it is only one automatically created object - the Project object - in the project pool. All other structures listed in the structure window are still empty.

Figure S1-3 shows the open Project Pool window with the Project object and the open Object window. The Attributes view of the Project Global Data aspect is opened and the corresponding dialog box presented for editing.


Structure Builder - [exp13r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
Close aspect area
exp13r0 (otmayer): Example Project EXP13R0
Aspects
New...
Edit...
Delete
More...
Views
Open
Help
Close
Set Attributes of Project Global Data Aspect
Customer
Responsible Company
OK
Cancel
Help

Example Project EXP13R0 Project

☑ Follow Object Selection

Description: Category (or Type):

Apparatus List	El. Report
Basic Property	Customer ID
Project Global Data	Project Global Data
Project OCS Defaults	Project OCS Defaults
Structure	+ Mounting Location Structure
Structure	++ Location Structure
Structure	= Function Structure

Views: Attributes

Customer

Name: EXP13R0
Project Name (1): Example Project
Project Name (2): AdvaBuild Engineering Software
Project Name (3):
Factory:
District:

Responsible Company

Name: ABB Automation Products
Department: APR/L
Project Leader: K. Auer
Revision: 0
Start Date: 11/19/97 10:37:59 AM

Figure S1-3. Editing Project Global Data Aspect

Step 2 Build Library Environment

An Object library project is delivered with the Advant Engineering Workplace in conjunction with an example project. The Object library has the product name:

Advant Object Library for Advant Controllers

It provides the basis by which the overall functionality of the AdvaBuild Engineering software can be verified. In the following we refer to this library as **Object library.**

The contents of the Object library project with the relevant expansion done by the user can be used in any project by applying import functionality or by copying objects from the library to a project.

The installation of the Object library and the Example Project is performed by using the Export/Import functionality of the Structure Builder.

S2.1 Installation of the Object Library Project

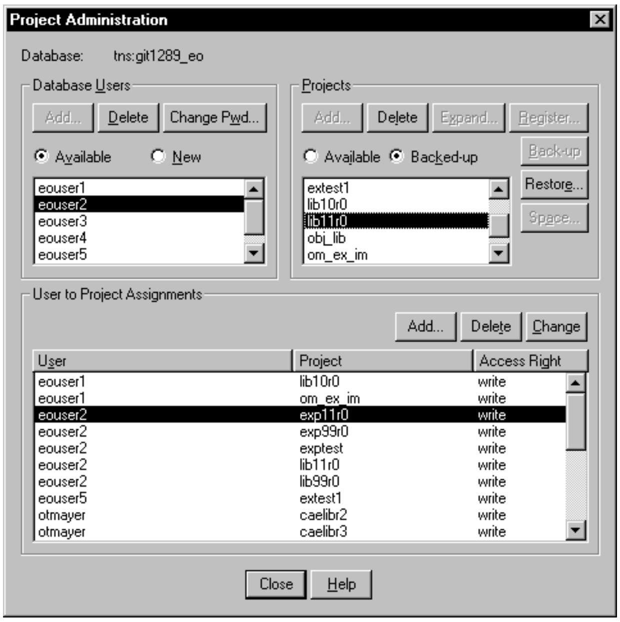
S2.1.1 Setup the File System from the CD

See the *Advant Object Library for Advant Controllers Release Notes* on how to setup the file system of the Object library to get it ready for import.

S2.1.2 Import the Object Library Project

The Object library is installed from a file system by using the Backup / Restore functionality.

Import the Object Library Project	
Preconditions:	File system of the Object library assigned to directory **advabld1\data\eobackup** of the server system
User activities:	1. Login to Windows NT as user belonging to the AdvaBuild engineers group (e.g. **eouser1**) 2. Open the Structure Builder 3. Choose **Administration** from the Project menu 4. Enter the connect string **tns:*servername*_eo** for the database and enter your administrator password in the **Open Database** box and click on **OK** 5. Click on **Backed-up** in the Projects group box in the **Project Administration** window 6. Click on **lib11r0** in the list of exported projects 7. Press **Restore** in the Projects group box in the **Project Administration** window 8. Enter the Restore Project Name in the **Restore Project** Window. (You can also use as default the source project name lib11r0) 9. Click on **OK** in the **Restore Project** window 10. Press **Add** in the User to Project Assignments group box of the **Project Administration** window and assign User(s) to the restored Object library
Result:	Object library restored and available for its use



Database: tns:git1289_eo

Database Users: ● Available ○ New

eouser1
eouser2
eouser3
eouser4
eouser5

Projects: ○ Available ● Backed-up

extest1
lib10r0
lib11r0
obj_lib
om_ex_im

User	Project	Access Right
eouser1	lib10r0	write
eouser1	om_ex_im	write
eouser2	exp11r0	write
eouser2	exp99r0	write
eouser2	exptest	write
eouser2	lib11r0	write
eouser2	lib99r0	write
eouser5	extest1	write
otmayer	caelibr2	write
otmayer	caelibr3	write

Figure S2-1. Project Administration Window Showing Exported Projects

S2.2 Structure and Content of the Object Library

NOTE

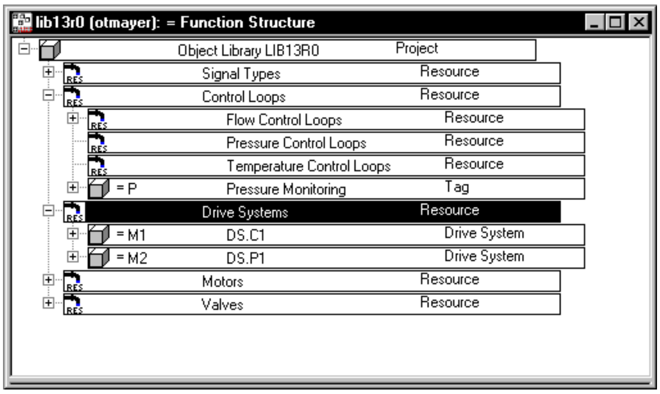
For a more detailed information than given below about the content of the Object library see the *Advant Object Library for Advant Controllers Reference Manual.*

S2.2.1 Function Oriented Structure

The object types related to this structure are:

- Control Loops
- Drive Systems
- Devices (in its functional context)
- Signals.

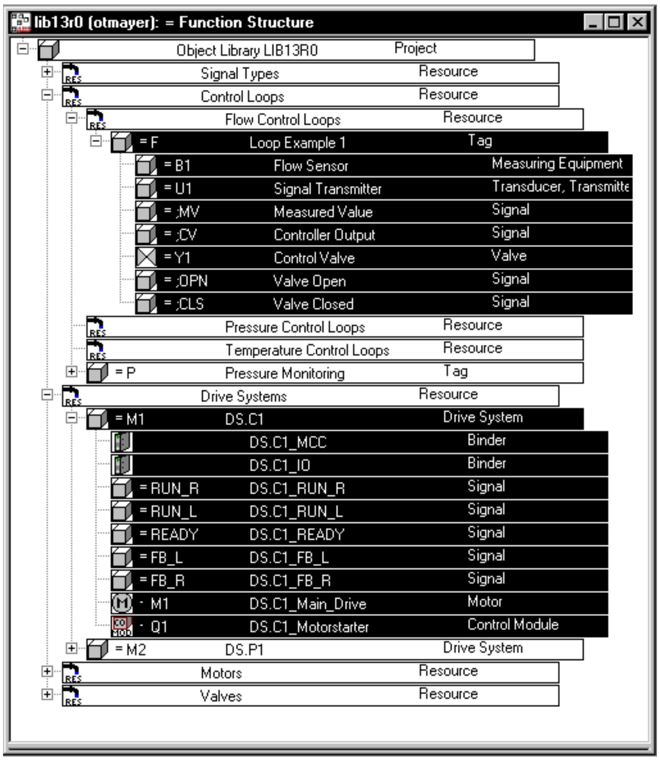
Figure S2-2 shows the basic Function Structure of the Object library project.



```
lib13r0 (otmayer): = Function Structure
Object Library LIB13R0                      Project
  Signal Types                              Resource
  Control Loops                             Resource
    Flow Control Loops                      Resource
    Pressure Control Loops                  Resource
    Temperature Control Loops               Resource
    = P   Pressure Monitoring               Tag
  Drive Systems                             Resource
    = M1  DS.C1                             Drive System
    = M2  DS.P1                             Drive System
  Motors                                    Resource
  Valves                                    Resource
```

Figure S2-2. Basic Structure of Function Items

Figure S2-3 shows the Advant objects of a control loop example and of a drive system example. These objects are placed in the Function Structure.



```
lib13r0 (otmayer): = Function Structure
Object Library LIB13R0                      Project
  Signal Types                              Resource
  Control Loops                             Resource
    Flow Control Loops                      Resource
      = F   Loop Example 1                  Tag
        = B1   Flow Sensor                  Measuring Equipment
        = U1   Signal Transmitter           Transducer, Transmitte
        = ;MV  Measured Value               Signal
        = ;CV  Controller Output            Signal
        = Y1   Control Valve                Valve
        = ;OPN Valve Open                   Signal
        = ;CLS Valve Closed                 Signal
    Pressure Control Loops                  Resource
    Temperature Control Loops               Resource
    = P   Pressure Monitoring               Tag
  Drive Systems                             Resource
    = M1  DS.C1                             Drive System
            DS.C1_MCC                       Binder
            DS.C1_IO                        Binder
      = RUN_R   DS.C1_RUN_R                 Signal
      = RUN_L   DS.C1_RUN_L                 Signal
      = READY   DS.C1_READY                 Signal
      = FB_L    DS.C1_FB_L                  Signal
      = FB_R    DS.C1_FB_R                  Signal
      - M1      DS.C1_Main_Drive            Motor
      - Q1      DS.C1_Motorstarter          Control Module
    = M2  DS.P1                             Drive System
  Motors                                    Resource
  Valves                                    Resource
```

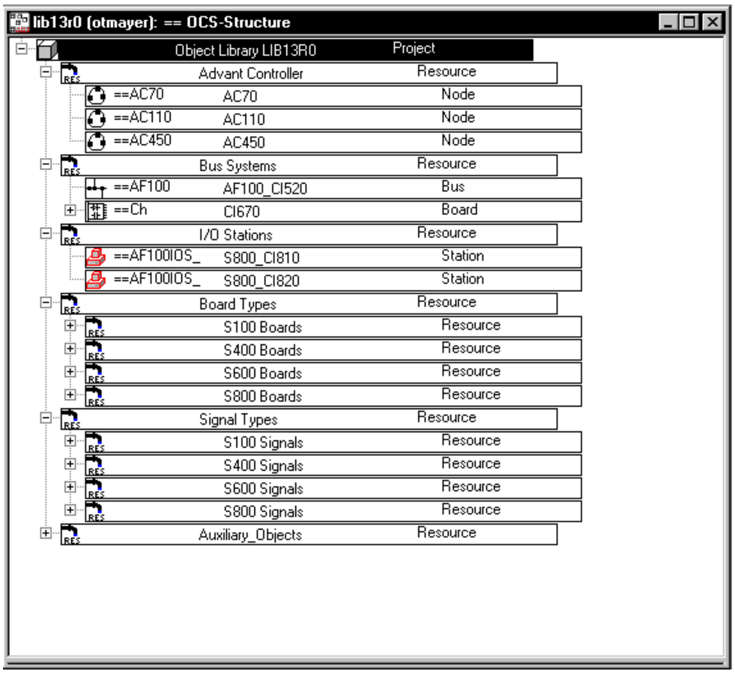
Figure S2-3. Example of a Control Loop and a Drive System in the Object Library

S2.2.2 Control System Oriented Structure

The following object types shall be provided in the Control System Structure (OCS-Structure) of the Object library project:

- Advant Controllers
- Bus Systems
- I/O Stations
- Boards
- Signals.

Figure S2-4 shows the basic Control System Structure of the Object library project.



lib13r0 (otmayer): == OCS-Structure

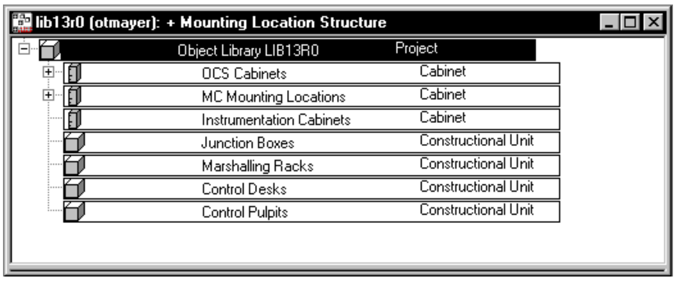
	Object Library LIB13R0	Project
	Advant Controller	Resource
==AC70	AC70	Node
==AC110	AC110	Node
==AC450	AC450	Node
	Bus Systems	Resource
==AF100	AF100_CI520	Bus
==Ch	CI670	Board
	I/O Stations	Resource
==AF100IOS_	S800_CI810	Station
==AF100IOS_	S800_CI820	Station
	Board Types	Resource
	S100 Boards	Resource
	S400 Boards	Resource
	S600 Boards	Resource
	S800 Boards	Resource
	Signal Types	Resource
	S100 Signals	Resource
	S400 Signals	Resource
	S600 Signals	Resource
	S800 Signals	Resource
	Auxiliary_Objects	Resource

Figure S2-4. Control System Structure in the Object Library

S2.2.3 Mounting Location Structure

Objects in the library that represent mounting locations, such as cabinets, desks, pulpits etc. are placed in the Mounting Location Structure of the Object library.

The basic structure of Figure S2-5 shows the upper levels structured according to the different types of constructional units that are used as locations for electrical and control equipment.

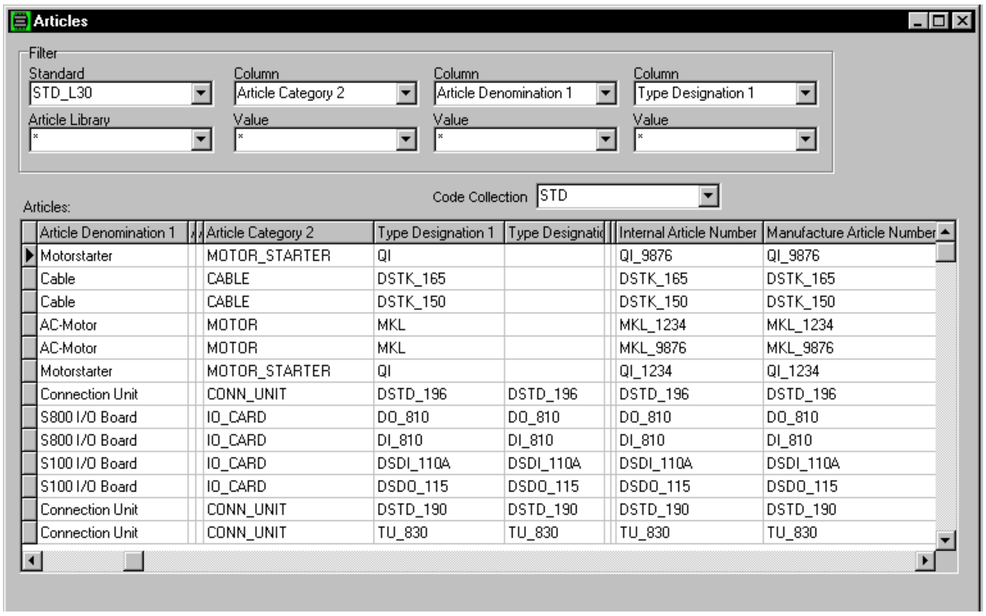


Object Library LIB13R0	Project
OCS Cabinets	Cabinet
MC Mounting Locations	Cabinet
Instrumentation Cabinets	Cabinet
Junction Boxes	Constructional Unit
Marshalling Racks	Constructional Unit
Control Desks	Constructional Unit
Control Pulpits	Constructional Unit

Figure S2-5. Basic Mounting Location Structure of the Object Library

S2.3 Assigning Articles to the Object Library

Articles with its relevant data for electrical engineering are assigned to a Standard. From Advant object aspects which represent articles is referenced to these article data in a Standard. An effective way to assign articles to a library project is to import it from the Standard of another project and to adapt it to the needs of the specific library project. The articles of an existing Standard have to be at first exported to a data file and from there it can be imported to the Standard of the library project. Figure S2-6 shows a datasheet view on a part of the articles assigned to a Standard.



Standard: STD_L30
Column: Article Category 2
Column: Article Denomination 1
Column: Type Designation 1
Article Library: *
Value: *
Value: *
Value: *
Code Collection: STD

Articles:

Article Denomination 1	Article Category 2	Type Designation 1	Type Designati	Internal Article Number	Manufacture Article Number
Motorstarter	MOTOR_STARTER	QI		QI_9876	QI_9876
Cable	CABLE	DSTK_165		DSTK_165	DSTK_165
Cable	CABLE	DSTK_150		DSTK_150	DSTK_150
AC-Motor	MOTOR	MKL		MKL_1234	MKL_1234
AC-Motor	MOTOR	MKL		MKL_9876	MKL_9876
Motorstarter	MOTOR_STARTER	QI		QI_1234	QI_1234
Connection Unit	CONN_UNIT	DSTD_196	DSTD_196	DSTD_196	DSTD_196
S800 I/O Board	IO_CARD	DO_810	DO_810	DO_810	DO_810
S800 I/O Board	IO_CARD	DI_810	DI_810	DI_810	DI_810
S100 I/O Board	IO_CARD	DSDI_110A	DSDI_110A	DSDI_110A	DSDI_110A
S100 I/O Board	IO_CARD	DSDO_115	DSDO_115	DSDO_115	DSDO_115
Connection Unit	CONN_UNIT	DSTD_190	DSTD_190	DSTD_190	DSTD_190
Connection Unit	CONN_UNIT	TU_830	TU_830	TU_830	TU_830

Figure S2-6. View on Contents of Articles Library

The procedure of exporting articles of an existing Standard to a data file and importing the articles from the data file to the Standard of the Object library is explained in the following.

Inspect and Export Articles from a Standard	
Preconditions:	Project with suitable Standard available
User activities:	1. Start up **elMaster Administrator** 2. Insert **Username** and **Password** of project with existing Standard 3. Inspect articles in existing Standard: Double Click icon **Articles** on the bottom of the **elMaster Administrator** window Select the Standard to be exported in the **Standard** list box within the **Filter** group box of the **Articles** window Set further filter criteria in the **Filter** group box of the **Articles** window, if relevant All articles that match the settings in the **Filter** group box are presented in the **Articles** table within in the **Articles** window 4. Export articles of selected Standard: Choose **Export** in the **File** menu of the **elMaster Administrator** window Choose a directory and insert filename for the exported library file in the **Save Article Export File** box (for example **…\tmp\art_lib**) and click on **Save** 5. Choose **Exit** from the **File** menu and quit **elMaster Administrator**
Result:	Articles of a Standard exported to data file

Import Articles from a Data File to a Standard	
Preconditions:	Data file as result of an article export available
User activities:	1. Start up **elMaster Administrator** 2. Insert **Username** and **Password** of the project to which a library shall be imported 3. Double Click on icon **Articles** on the bottom of **elMaster Administrator** window 4. Choose **Import** in the **File** menu of **elMaster Administrator** window 5. Select the exported article file you want to import in the **Import** box (for example **…\tmp\art_lib**) and click on **Open** 6. Select the option **Semicolon delimited text** in the **Import type** box and click on OK 7. Inspect the imported articles: Select the Standard to which was imported in the **Standard** list box within the **Filter** group box of the **Articles** window Set further filter criteria in the **Filter** group box of the **Articles** window, if relevant All articles that match the settings in the **Filter** group box are presented in the **Articles** table within in the **Articles** window 8. Choose **Exit** from the **File** menu and quit **elMaster Administrator**.
Result:	Articles assigned to the Standard of a project

S2.3.1 Creating the Articles in the Object Library

Articles are created/edited in an existing Standard by editing it manually or by copying article data into the articles data sheet. See the corresponding MS Access functionality.

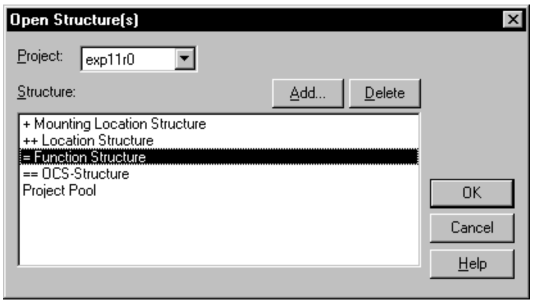
Step 3 Build the Basic Function Oriented Structure

S3.1 Opening the Function Oriented Structure

Start building up the Function Oriented Structure. Open the window of the Function Oriented Structure.

Opening a Structure Window	
Preconditions:	Project open
User activities:	1. Use the Structure Builder 2. Choose **Open Structure** in the **Window** menu 3. Select **Function Structure** in the **Open Structure** window 4. Click on **Open**
Result:	Function Structure window opens

Figure S3-1 shows the Open Structure window from where the structure to be opened is selected. You can select more than one structure.


Open Structure(s)

Project: exp11r0

Structure:

+ Mounting Location Structure
++ Location Structure
= Function Structure
== OCS-Structure
Project Pool

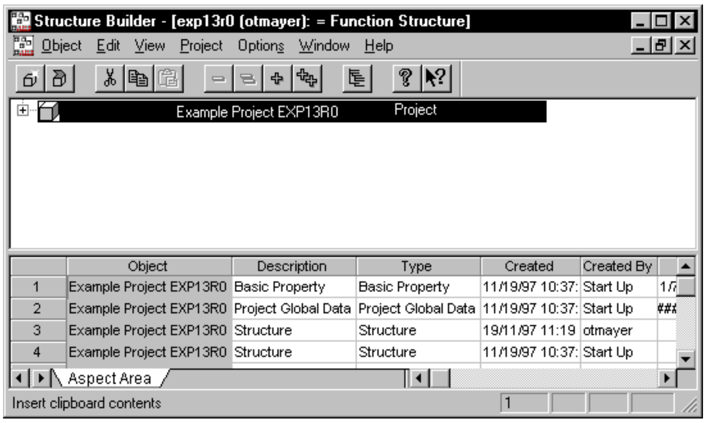
Figure S3-1. Opening the Function Structure in the Open Structure Window

S3.2 Placing the Project Object in the Function Oriented Structure

The Function Structure window is still empty. The top object in this structure shall be the project object. This object has already been created and is located in the project pool. Place this object into the Function Oriented Structure.

Placing the Project Object as Top Object of a Structure	
Preconditions:	Project created and project pool open
User activities:	1. Click on the **Project** object in the **Project Pool** window 2. Drag the **Project** object by pressing the right mouse button and drop it into the **Function Structure** window.
Result:	Project object is root object in the Function Structure

Figure S3-2 shows the Function Structure window with the placed project object. The object area and the aspect area are shown in this window.


Structure Builder - [exp13r0 (otmayer): = Function Structure]
Object
Edit
View
Project
Options
Window
Help
Example Project EXP13R0
Project
Aspect Area
Insert clipboard contents
1

	Object	Description	Type	Created	Created By	
1	Example Project EXP13R0	Basic Property	Basic Property	11/19/97 10:37:	Start Up	1/7
2	Example Project EXP13R0	Project Global Data	Project Global Data	11/19/97 10:37:	Start Up	###
3	Example Project EXP13R0	Structure	Structure	19/11/97 11:19	otmayer	
4	Example Project EXP13R0	Structure	Structure	11/19/97 10:37:	Start Up	

Figure S3-2. Function Structure Window with Placed Project Object

S3.3 Creation of Advant Objects in the Function Oriented Structure

The plant functions to be placed in the first level below the project are the functions of the main plant sections (subplants) which, in our example, are:

- Tundish
- Casting machine
- Torch cutting machine
- Ladle tower

- Conveyor system.

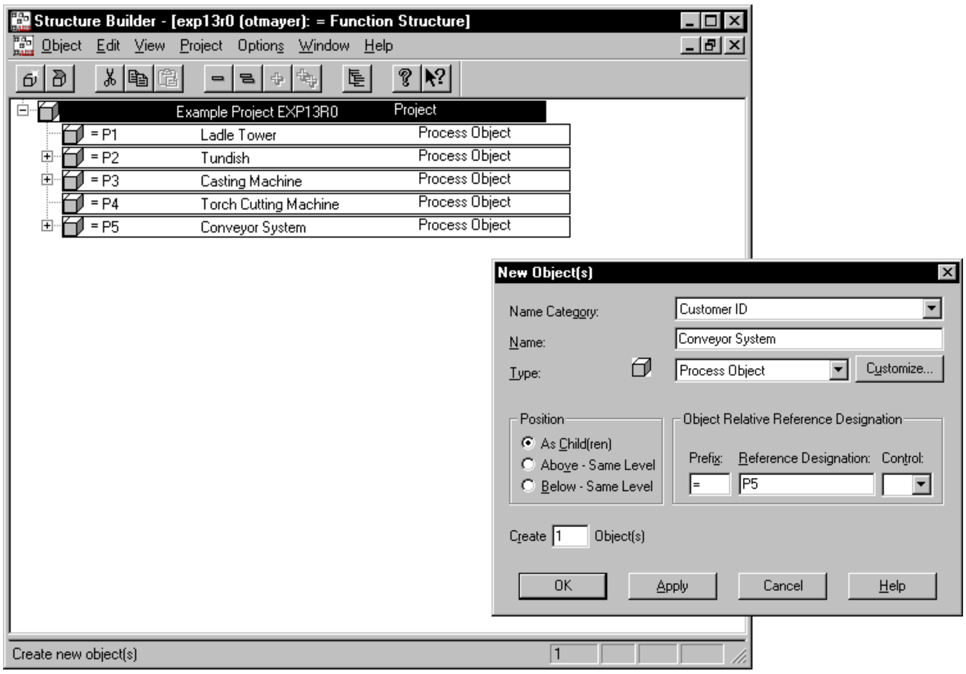
The Advant objects of interest in our engineering example fulfill functions that belong to the casting machine. Functions that control the casting machine are:

- Mould level control
- Mould cooling
- Dummy bar handling
- Strand drive system
- Roller cooling
- Spray cooling.

Create these objects in the Function Oriented Structure.

Creation of an Object in a Structure Window	
Preconditions:	Structure window open
User activities:	1. Use the Structure Builder 2. Click on the Advant object in the open structure to which a new objects shall be created either as child or above/below the selected object on same level. 3. Choose **New** in the **Object** menu 4. Select in the combo boxes of the New Objects window the – Category of Object Name (default is Customer Id.) – Object Type 5. Fill in the Object Name 6. Select one of the placing items – as child – above on same level – below on same level 7. Click on **OK**
Result:	New Advant object is created and placed in the selected structure and in the project pool

Figure S3-3 shows the Function Structure window with the New Objects window as presented for the creation of an object.


Structure Builder - [exp13r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
New Object(s)
Customize...
OK
Apply
Cancel
Help

Example Project EXP13R0 Project
= P1 Ladle Tower Process Object
= P2 Tundish Process Object
= P3 Casting Machine Process Object
= P4 Torch Cutting Machine Process Object
= P5 Conveyor System Process Object

Name Category: Customer ID
Name: Conveyor System
Type: Process Object

Position
● As Child(ren)
○ Above - Same Level
○ Below - Same Level

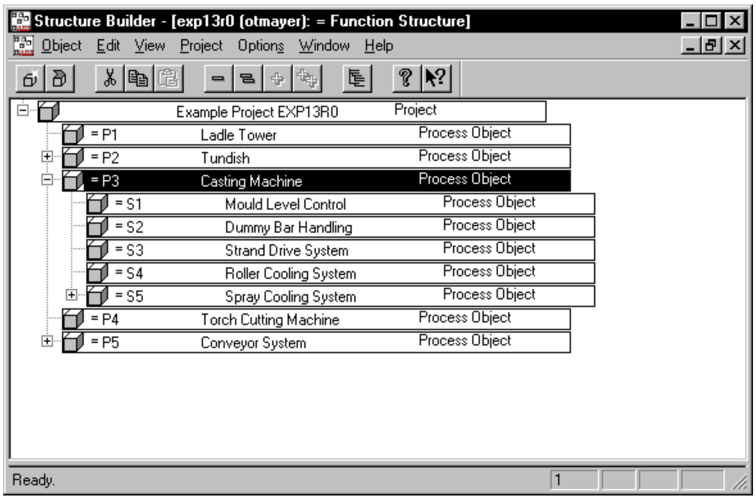
Object Relative Reference Designation
Prefix: =
Reference Designation: P5
Control:

Create 1 Object(s)

Create new object(s)

Figure S3-3. Creation of a New Object

Figure S3-4 shows the Function Oriented Structure after creation of all the aforementioned objects which represent the higher levels of the Function Structure.

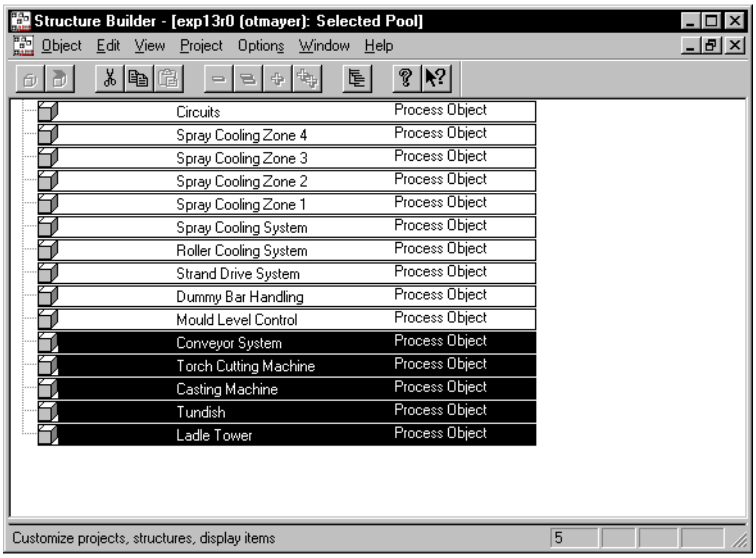

Structure Builder - [exp13r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help

Example Project EXP13R0 Project
= P1 Ladle Tower Process Object
= P2 Tundish Process Object
= P3 Casting Machine Process Object
= S1 Mould Level Control Process Object
= S2 Dummy Bar Handling Process Object
= S3 Strand Drive System Process Object
= S4 Roller Cooling System Process Object
= S5 Spray Cooling System Process Object
= P4 Torch Cutting Machine Process Object
= P5 Conveyor System Process Object

Ready.

Figure S3-4. Higher Levels of the Function Structure

All new created objects are also put into the project pool. Figure S3-5 shows the same objects located in the project pool as presented by the Project Pool window. (This figure also shows other objects created later in the engineering procedure for the example project).



Circuits	Process Object
Spray Cooling Zone 4	Process Object
Spray Cooling Zone 3	Process Object
Spray Cooling Zone 2	Process Object
Spray Cooling Zone 1	Process Object
Spray Cooling System	Process Object
Roller Cooling System	Process Object
Strand Drive System	Process Object
Dummy Bar Handling	Process Object
Mould Level Control	Process Object
Conveyor System	Process Object
Torch Cutting Machine	Process Object
Casting Machine	Process Object
Tundish	Process Object
Ladle Tower	Process Object

Customize projects, structures, display items 5

Figure S3-5. Project Pool

S3.4 Assigning Engineering Data to Advant Objects in the Basic Structure

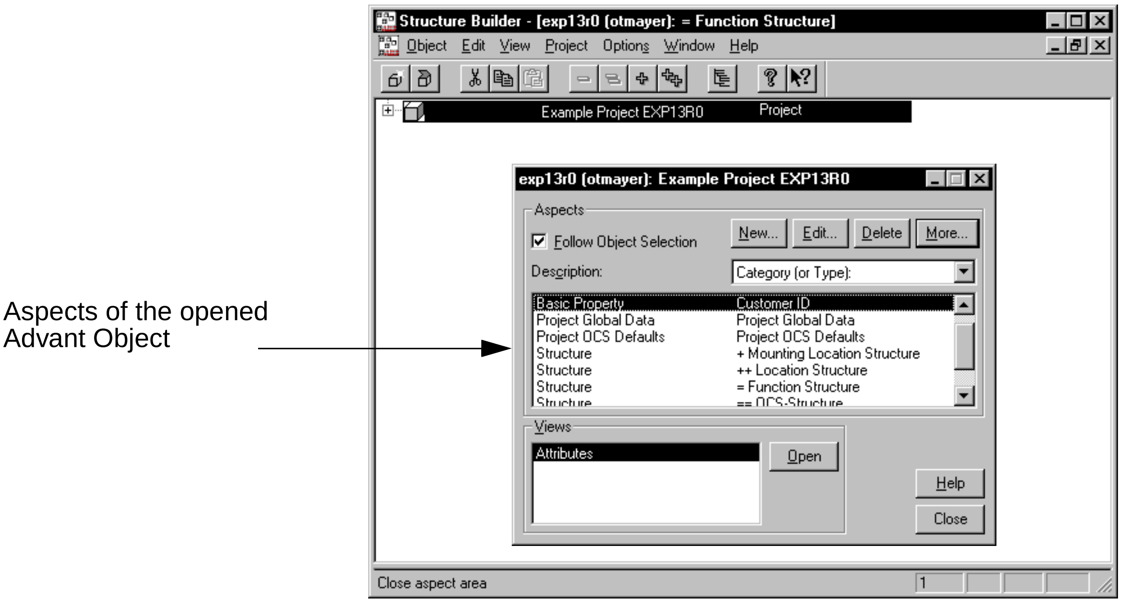
Engineering data are assigned to aspects of Advant objects. The data of an aspect is handled by a corresponding AdvaBuild Engineering Software product. You can edit aspect data after first opening an Advant object and then one of the predefined views on the aspect.

S3.4.1 Opening an Advant Object

When opening an Advant object, all its aspects are presented in the Object window of the Advant object.

Opening an Advant Object	
User activities:	1. Use the Structure Builder 2. Click on the Advant Object to be opened 3. Choose **Open** in the **Object** menu
Result:	The Aspect window of the selected object is open and presents the assigned aspects

Figure S3-6 shows the Object window of the Project object of our example.



Structure Builder - [exp13r0 (otmayer): = Function Structure]

Example Project EXP13R0 Project

exp13r0 (otmayer): Example Project EXP13R0

☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
Project Global Data	Project Global Data
Project OCS Defaults	Project OCS Defaults
Structure	+ Mounting Location Structure
Structure	++ Location Structure
Structure	= Function Structure
Structure	== OCS-Structure

Attributes

Close aspect area 1

Figure S3-6. Project Object Window

S3.4.2 Creation of Aspects

S3.4.2.1 Automatically Created Aspects

Figure S3-6 shows several aspects in the Object window. Some of these aspects are created automatically when an object is created or placed in structures. Such aspects are:

- Basic Properties Aspect
- Structure Aspects (information in which structure an object is placed)
- Project Global Data (only in the project object)
- Project OCS Defaults (only in the project object).

The Basic Properties Aspect is automatically created when an Advant object is created. It is a container for names or identities of the object. Several names or identities may be assigned via this aspect type to an Advant object. The data can be filled into the following predefined attributes:

- Customer Id
- Document Id
- Function Id
- Instance Id

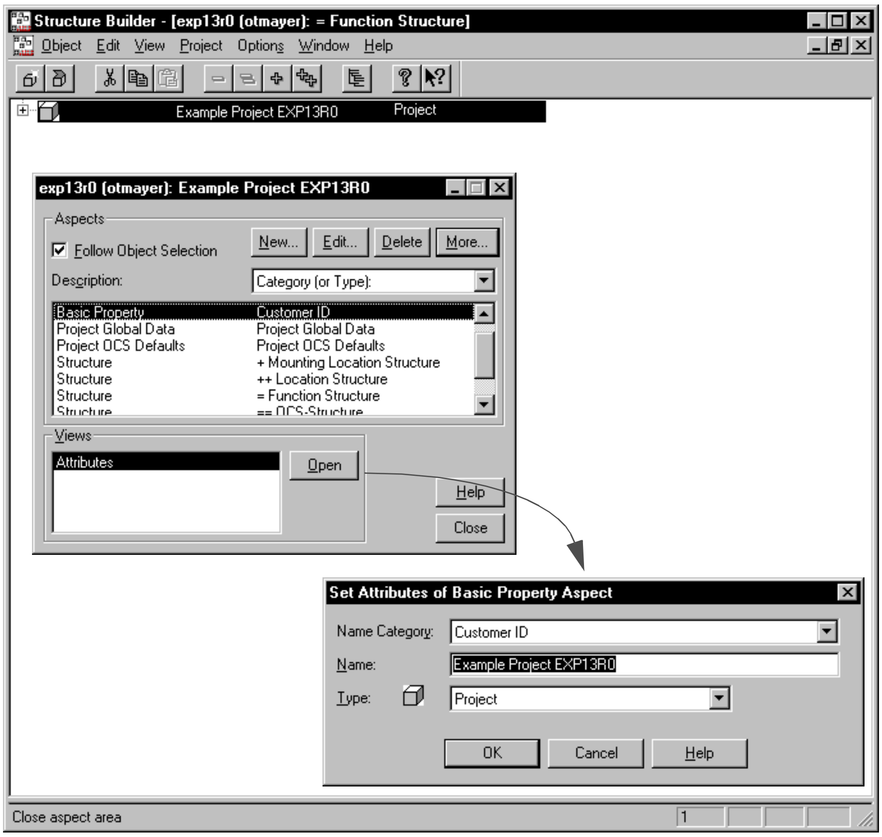
- Instrumentation Id
- Location Id
- Mechanical Id
- Process Id
- Product Id.

One of these Attributes is presented on the object display. You can select which one. Customer Id is default.

The Structure aspects contain the information about the placement of an Advant object in a structure. Opening the Structure view of this aspect view presents the Advant object in the corresponding Structure window.

The Project Global Data Aspect is an aspect assigned only to the project object. It contains some basic project data.

Figure S3-7 shows the Object window of the project object with the selected Basic Properties aspect and the Set Attributes window.



Structure Builder - [exp13r0 (otmayer): = Function Structure]

Example Project EXP13R0 Project

exp13r0 (otmayer): Example Project EXP13R0

Aspects

☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
Project Global Data	Project Global Data
Project OCS Defaults	Project OCS Defaults
Structure	+ Mounting Location Structure
Structure	++ Location Structure
Structure	= Function Structure
Structure	== OCS-Structure

Views

Attributes

Set Attributes of Basic Property Aspect

Name Category: Customer ID

Name: Example Project EXP13R0

Type: Project

Close aspect area

Figure S3-7. Editing Attributes of Basic Properties

S3.4.2.2 User Added Aspects

All aspects of an Advant object must be created by the user, except the aforementioned in Section 3.4.2.1. There are several aspect types predefined. Tools are assigned to each aspect type that handles the aspect data.

Adding an Aspect to an Advant Object	
Preconditions:	Advant object created
User activities:	1. Use the Structure Builder 2. Click on the Advant object (for example in the Function Structure) to which an aspect shall be added. 3. Open the ***Object*** window 4. Click on **New** 5. Select the Aspect Type in the **New Aspect** window 6. Insert the Aspect Description in the **New Aspect** window 7. Click on **OK**
Result:	The selected aspect type is added to the Advant object

Figure S3-8 shows the windows involved when a new aspect is added to an Advant object.

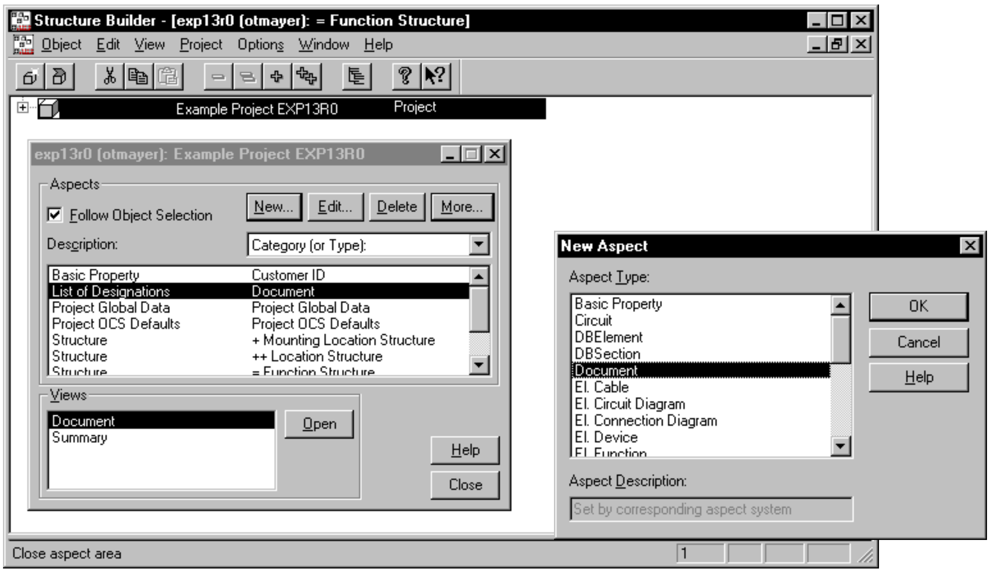

Structure Builder - [exp13r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
Example Project EXP13R0 Project
exp13r0 (otmayer): Example Project EXP13R0
Aspects
☑ Follow Object Selection
New... Edit... Delete More...
Description:
Category (or Type):
Basic Property Customer ID
List of Designations Document
Project Global Data Project Global Data
Project OCS Defaults Project OCS Defaults
Structure + Mounting Location Structure
Structure ++ Location Structure
Structure = Function Structure
Views
Document
Summary
Open
Help
Close
New Aspect
Aspect Type:
Basic Property
Circuit
DBElement
DBSection
Document
El. Cable
El. Circuit Diagram
El. Connection Diagram
El. Device
El. Function
OK
Cancel
Help
Aspect Description:
Set by corresponding aspect system
Close aspect area
1

Figure S3-8. Adding an Aspect to an Advant Object

By opening one of the views on an aspect, you get access to the tool by which the aspect data can be edited.

Continue with our example project. The information used as basis for the creation and structuring of the upper two hierarchical levels in the Function Structure, was taken out from the plant and process describing documents delivered by the customer or process partner. These documents should be assigned as aspects to those Advant objects which are subject of the relevant document. Back to our example, have a look at the project object.

The access to data describing the Advant object is via the aspects of an object. Data that concern the project as a whole can be assigned to the project object. This means that the aspects containing these data are assigned to the project object. Such aspects are:

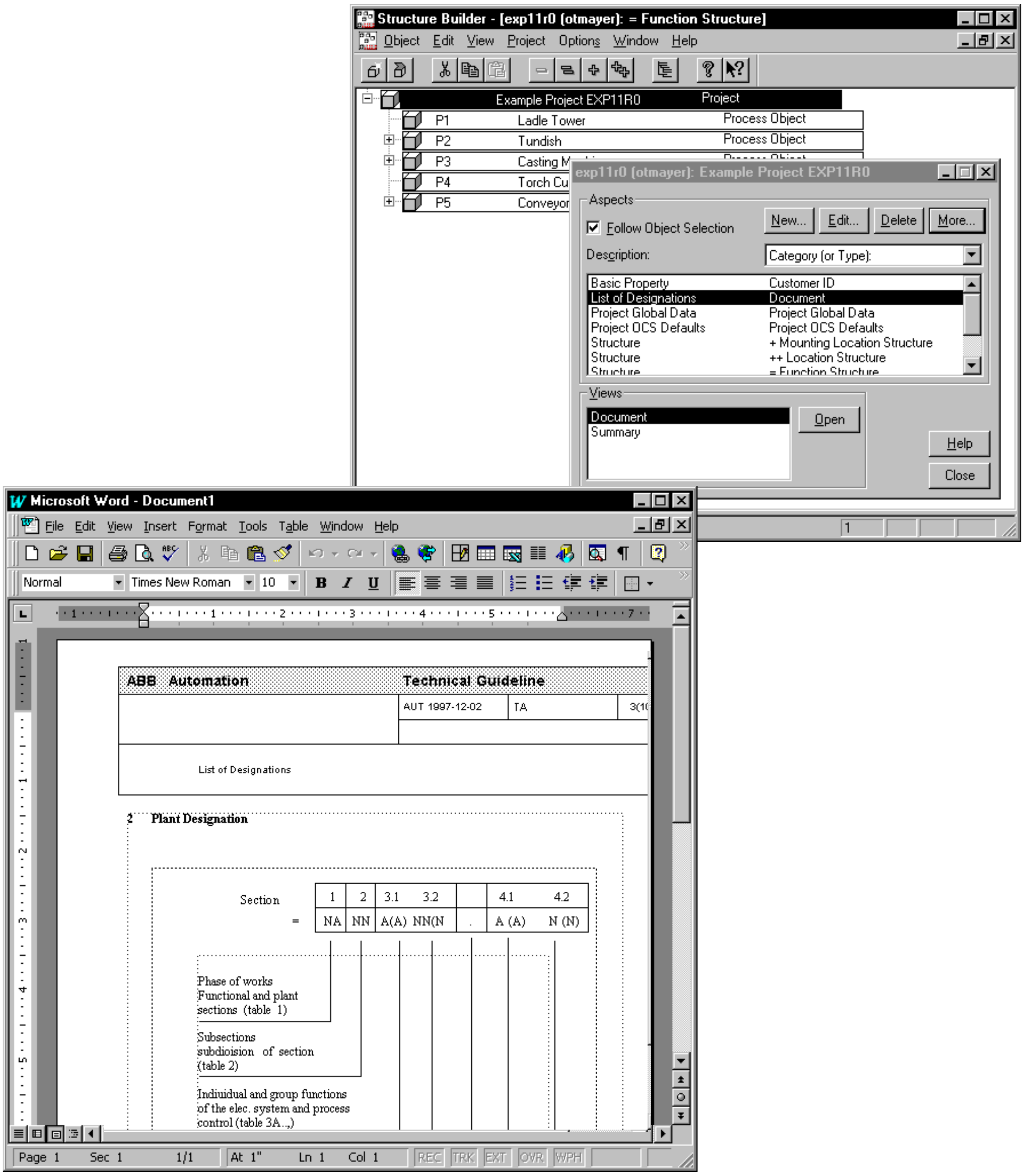
- Contract
- Scope of supply
- Description of the Designation System for the plant.

If you want to handle the aspects of an object, open this object. You have then access to the aspects and can open the aspect views which present the aspect data.

S3.4.2.3 Editing Aspect Data

Opening a View on an Aspect for Inserting, Modifying or Inspecting Data	
Preconditions:	Aspect to Advant object assigned
User activities:	1. Use the Structure Builder 2. Click on the Advant object (for example in the Function Structure) to which an aspect shall be added. 3. Open the ***Object*** window 4. Click on the aspect of interest in the ***Object*** window (for example aspect List of Designations of type Document) 5. Select the view on the aspect data in the ***Object*** window and click on **Open** (for example select Document view to open a document)
Result:	View on the aspect data is presented by the tool that handles the aspect view.

Figure S3-9 shows the windows involved when a view on an aspect is opened. This example shows opening the Document view of the aspect List of Designation of type Document.


Structure Builder - [exp11r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
exp11r0 (otmayer): Example Project EXP11R0
Aspects
New... Edit... Delete More...
Views
Open
Help
Close
Microsoft Word - Document1
File Edit View Insert Format Tools Table Window Help
Normal
Times New Roman
10
Page 1 Sec 1 1/1 At 1" Ln 1 Col 1 REC TRK EXT OVR WPH

	Example Project EXP11R0	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
P3	Casting M	
P4	Torch Cu	
P5	Conveyor	

☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
List of Designations	Document
Project Global Data	Project Global Data
Project OCS Defaults	Project OCS Defaults
Structure	+ Mounting Location Structure
Structure	++ Location Structure
Structure	= Function Structure

Document
Summary

ABB Automation **Technical Guideline**

AUT 1997-12-02 TA 3(1

List of Designations

2 Plant Designation

Section	1	2	3.1	3.2		4.1	4.2
=	NA	NN	A(A)	NN(N	.	A (A)	N (N)

Phase of works
Functional and plant sections (table 1)

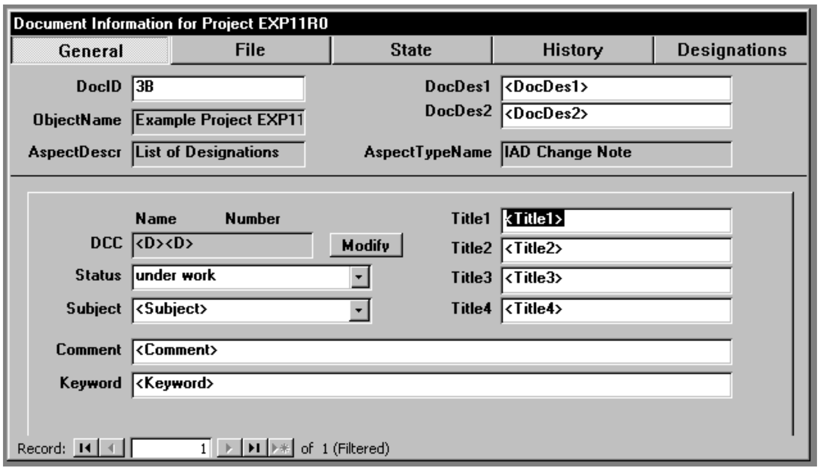
Subsections
subdioision of section
(table 2)

Indiuidual and group functions
of the elec. system and process
control (table 3A...)

Figure S3-9. Opening a View on a Document Aspect

Aspects representing descriptions and diagrams have besides the document view also an additional “Summary” view defined in their aspect types. By this view you can edit the footer attributes of the document and insert short descriptions.

Figure S3-10 shows an example of a Summary Form view of an aspect which represents a document. The data is handled by MS Access. Some of the data correspond to footer data of the document. Such footer data is automatically updated in the documents.



DocID 3B
ObjectName Example Project EXP11
AspectDescr List of Designations
DocDes1 <DocDes1>
DocDes2 <DocDes2>
AspectTypeName IAD Change Note
DCC <D><D>
Status under work
Subject <Subject>
Title1 <Title1>
Title2 <Title2>
Title3 <Title3>
Title4 <Title4>
Comment <Comment>
Keyword <Keyword>

Figure S3-10. View on the Summary Form of a Document Aspect

The objects created in our example on the level below the Project object are specified by aspects representing Descriptions and Diagrams. These documents specify the process functions of the plant part represented by the objects and also the requirements for the electrical and control system.

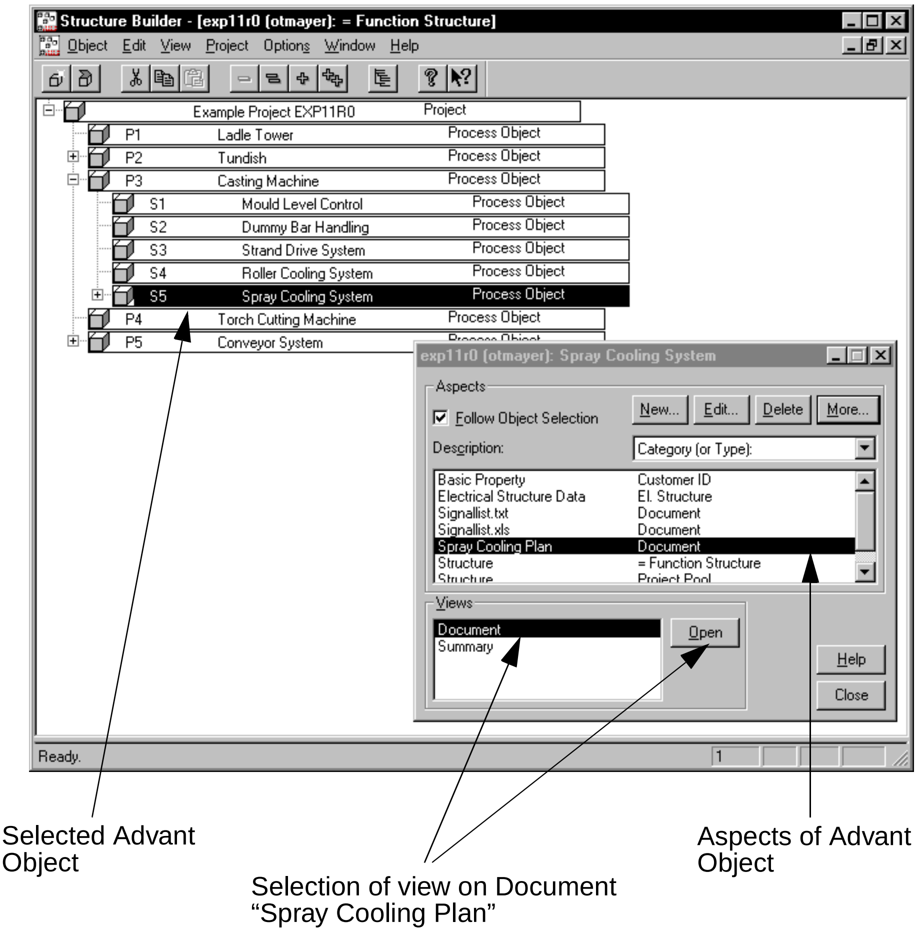
Typical aspects on this level represent:

- Process diagrams
- P&I Diagrams
- Process Descriptions
- Function Descriptions.

The customer documents describing the requirements on the spray cooling system are assigned to the Advant object that represents the spray cooling system.

For example the P&I-Diagram of the spray cooling system is a functional specification of the spray cooling system. The data of the P&I diagram are therefore put into the “P&I-Diagram aspect” of the Advant object “Spray Cooling System”.

Figure S3-11 shows the Object window of the Advant object that represents the *Strand Spray Cooling System*. In its Document aspect it contains the document *Spray Cooling Plan* as part of a functional description which could have been delivered by the customer.


Structure Builder - [exp11r0 (otmayer): = Function Structure]
Object
Edit
View
Project
Options
Window
Help
Ready.
1
exp11r0 (otmayer): Spray Cooling System
Aspects
New...
Edit...
Delete
More...
Views
Open
Help
Close

	Example Project EXP11R0	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
P3	Casting Machine	Process Object
S1	Mould Level Control	Process Object
S2	Dummy Bar Handling	Process Object
S3	Strand Drive System	Process Object
S4	Roller Cooling System	Process Object
S5	Spray Cooling System	Process Object
P4	Torch Cutting Machine	Process Object
P5	Conveyor System	Process Object

☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
Electrical Structure Data	El. Structure
Signallist.txt	Document
Signallist.xls	Document
Spray Cooling Plan	Document
Structure	= Function Structure
Structure	Project Pool

Views: Document, Summary

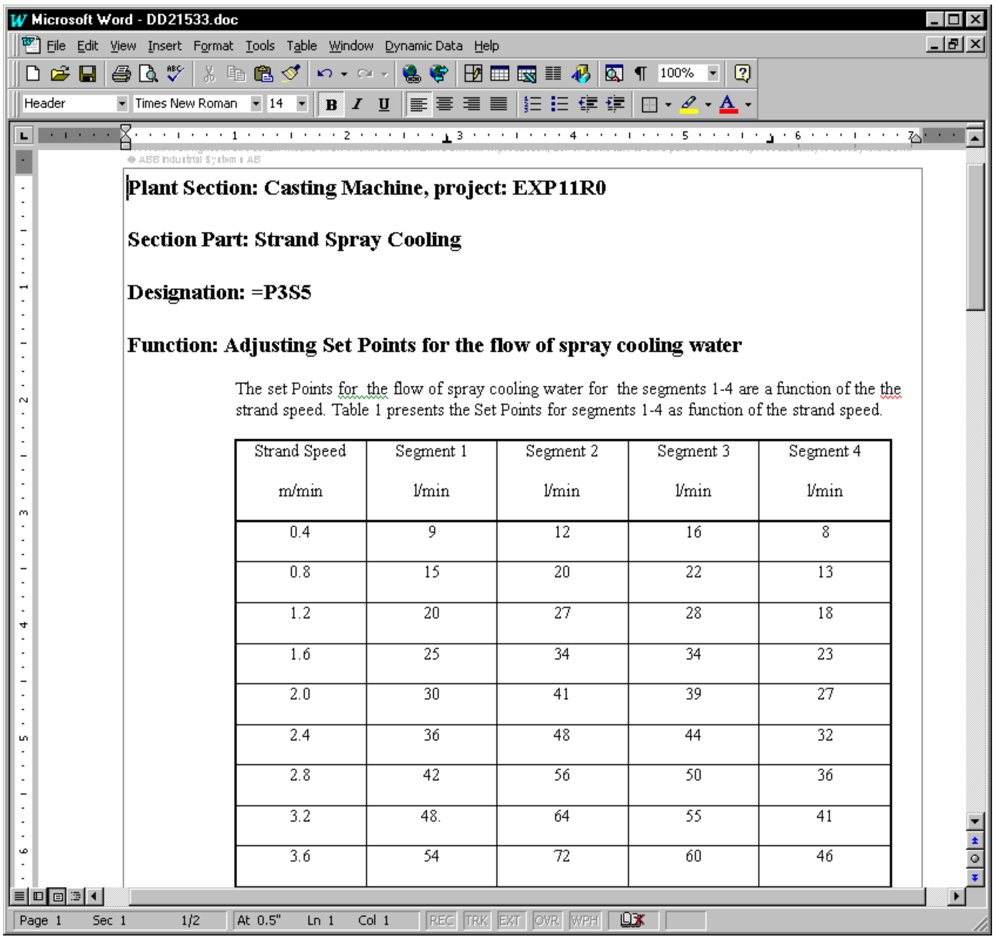
Selected Advant Object

Selection of view on Document "Spray Cooling Plan"

Aspects of Advant Object

Figure S3-11. Document Aspect of the Advant Object "Spray Cooling System"

Figure S3-12 shows the view on the document "Spray Cooling Plan" presented in MS Word.



Microsoft Word - DD21533.doc
File Edit View Insert Format Tools Table Window Dynamic Data Help
Header
Times New Roman
14
100%
Page 1 Sec 1 1/2 At 0.5" Ln 1 Col 1 REC TRK EXT OVR WPH

Plant Section: Casting Machine, project: EXP11R0

Section Part: Strand Spray Cooling

Designation: =P3S5

Function: Adjusting Set Points for the flow of spray cooling water

The set Points for the flow of spray cooling water for the segments 1-4 are a function of the the strand speed. Table 1 presents the Set Points for segments 1-4 as function of the strand speed.

Strand Speed m/min	Segment 1 l/min	Segment 2 l/min	Segment 3 l/min	Segment 4 l/min
0.4	9	12	16	8
0.8	15	20	22	13
1.2	20	27	28	18
1.6	25	34	34	23
2.0	30	41	39	27
2.4	36	48	44	32
2.8	42	56	50	36
3.2	48.	64	55	41
3.6	54	72	60	46

Figure S3-12. Document View Presented in MS Word

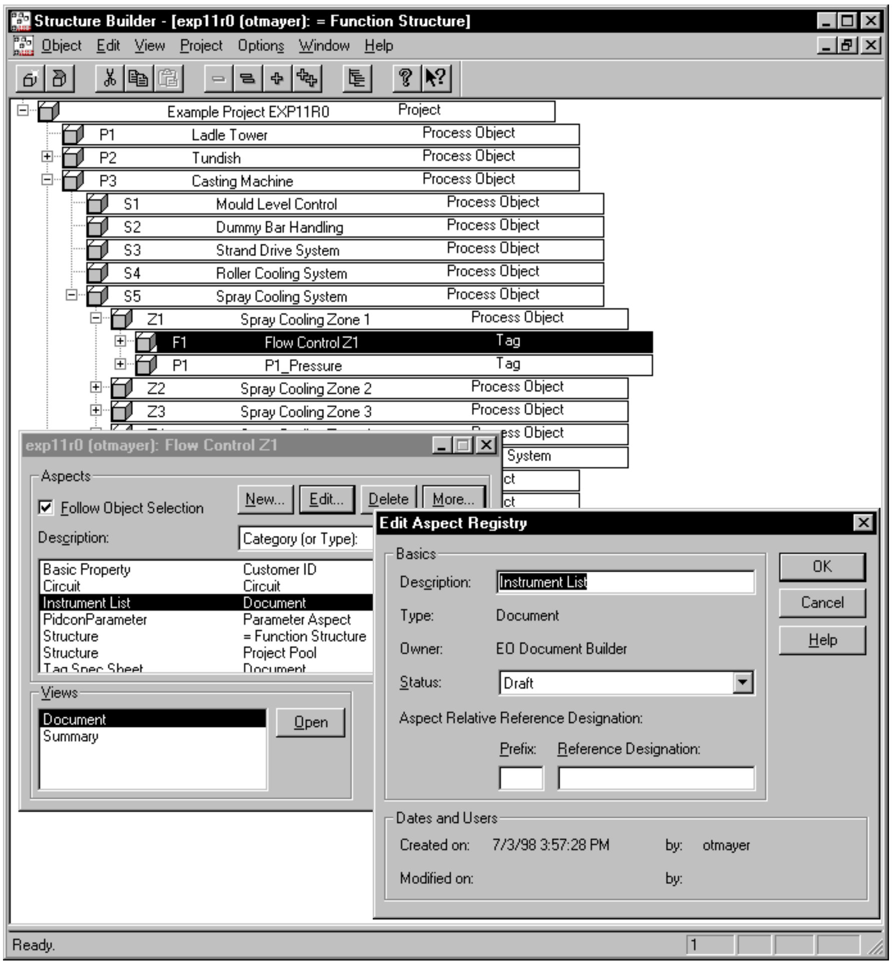
S3.4.2.4 Aspect Description and Administration

Aspect description and administration data as creation date, modification date, access rights, version, status, etc., are presented in a template for inspection and editing. Two types of aspect presentations are available:

- Aspect Registry box for a single object
- Aspect List presentation of all aspects of an object.

.

Opening the Edit Aspect Registry Box	
User activities:	1. Use the Structure Builder 2. Select the Aspect in the ***Object*** window for which the **Edit Aspect Registry** box shall be opened 3. Click on **Edit** in the ***Object*** window
Result:	Edit Aspect Registry box appears for editing


Structure Builder - [exp11r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
exp11r0 (otmayer): Flow Control Z1
Edit Aspect Registry
OK
Cancel
Help
New...
Edit...
Delete
More...
Open
Ready.
1

	Example Project EXP11R0	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
P3	Casting Machine	Process Object
S1	Mould Level Control	Process Object
S2	Dummy Bar Handling	Process Object
S3	Strand Drive System	Process Object
S4	Roller Cooling System	Process Object
S5	Spray Cooling System	Process Object
Z1	Spray Cooling Zone 1	Process Object
F1	Flow Control Z1	Tag
P1	P1_Pressure	Tag
Z2	Spray Cooling Zone 2	Process Object
Z3	Spray Cooling Zone 3	Process Object

Aspects

☑ Follow Object Selection

Description:	Category (or Type):
Basic Property	Customer ID
Circuit	Circuit
Instrument List	Document
PidconParameter	Parameter Aspect
Structure	= Function Structure
Structure	Project Pool
Tag Spec Sheet	Document

Views

Document
Summary

Basics

Description: Instrument List

Type: Document

Owner: EO Document Builder

Status: Draft

Aspect Relative Reference Designation:

Prefix: Reference Designation:

Dates and Users

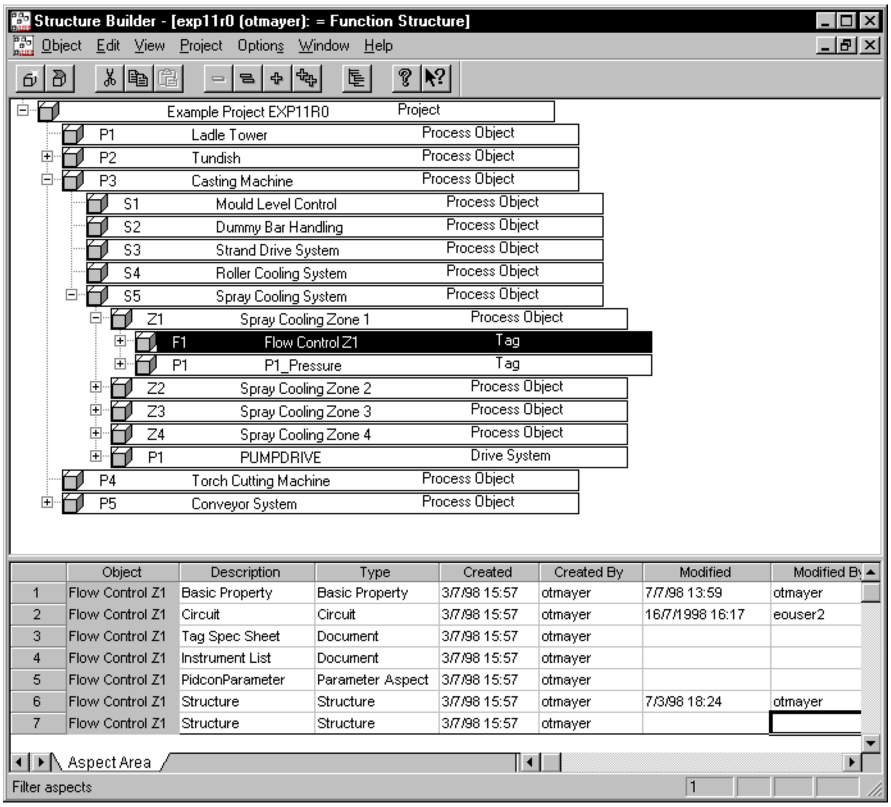
Created on: 7/3/98 3:57:28 PM by: otmayer

Modified on: by:

Figure S3-13. Opening the Aspect Registry Box

Figure S3-13 shows the selected aspect in the Object window and its open Edit Aspect Registry box.

Figure S3-14 shows the list presentation of all aspects of an Advant object representing a tag in our example.



Structure Builder - [exp11r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
Aspect Area
Filter aspects
1

- Example Project EXP11R0 Project
  - P1 Ladle Tower Process Object
  - P2 Tundish Process Object
  - P3 Casting Machine Process Object
    - S1 Mould Level Control Process Object
    - S2 Dummy Bar Handling Process Object
    - S3 Strand Drive System Process Object
    - S4 Roller Cooling System Process Object
    - S5 Spray Cooling System Process Object
      - Z1 Spray Cooling Zone 1 Process Object
        - F1 Flow Control Z1 Tag
        - P1 P1_Pressure Tag
      - Z2 Spray Cooling Zone 2 Process Object
      - Z3 Spray Cooling Zone 3 Process Object
      - Z4 Spray Cooling Zone 4 Process Object
      - P1 PUMPDRIVE Drive System
  - P4 Torch Cutting Machine Process Object
  - P5 Conveyor System Process Object

	Object	Description	Type	Created	Created By	Modified	Modified B
1	Flow Control Z1	Basic Property	Basic Property	3/7/98 15:57	otmayer	7/7/98 13:59	otmayer
2	Flow Control Z1	Circuit	Circuit	3/7/98 15:57	otmayer	16/7/1998 16:17	eouser2
3	Flow Control Z1	Tag Spec Sheet	Document	3/7/98 15:57	otmayer		
4	Flow Control Z1	Instrument List	Document	3/7/98 15:57	otmayer		
5	Flow Control Z1	PidconParameter	Parameter Aspect	3/7/98 15:57	otmayer		
6	Flow Control Z1	Structure	Structure	3/7/98 15:57	otmayer	7/3/98 18:24	otmayer
7	Flow Control Z1	Structure	Structure	3/7/98 15:57	otmayer		

Figure S3-14. Structure Window with Object Area and Aspect Area

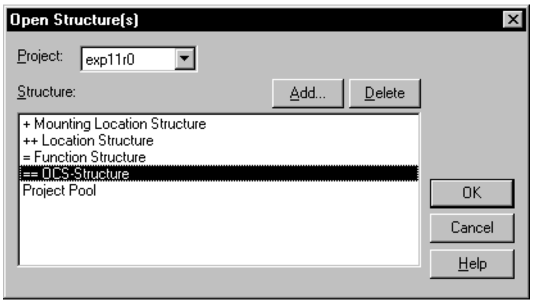
Step 4 Build the Basic OCS-Structure

S4.1 Opening the Control System Structure

Start building up the Control System Structure. Open the window of the Control System Structure.

Opening a Structure Window	
Preconditions:	Project open
User activities:	1. Use the Structure Builder 2. Choose **Open Structure** in the **Window** menu 3. Select **Control System Structure** in the **Open Structure** window 4. Click on **Open**
Result:	Control System Structure window opens

Figure S4-1 shows the Project Structure window from where the structure to be opened is selected. You can select more than one structure.



Project: exp11r0

+ Mounting Location Structure
++ Location Structure
= Function Structure
== OCS-Structure
Project Pool

Figure S4-1. Opening the OCS-Structure in the Open Structure Window

S4.2 Placing the Project Object in the Control System Structure

The Control System Structure window is still empty. The top object in this structure shall be the project object. This object has already been created and is located in the project pool. Place this object into the Control System Structure.

Placing the Project Object as Top Object of a Structure	
Preconditions:	Project created and opened
User activities:	1. Click on the **Project** object in the **Project Pool** window 2. Drag the **Project** object by pressing the right mouse button and drop it into the **Control System Structure** window.
Result:	Project object is root object in the Control System Structure

S4.3 Creating Objects in the Basic Control System Structure

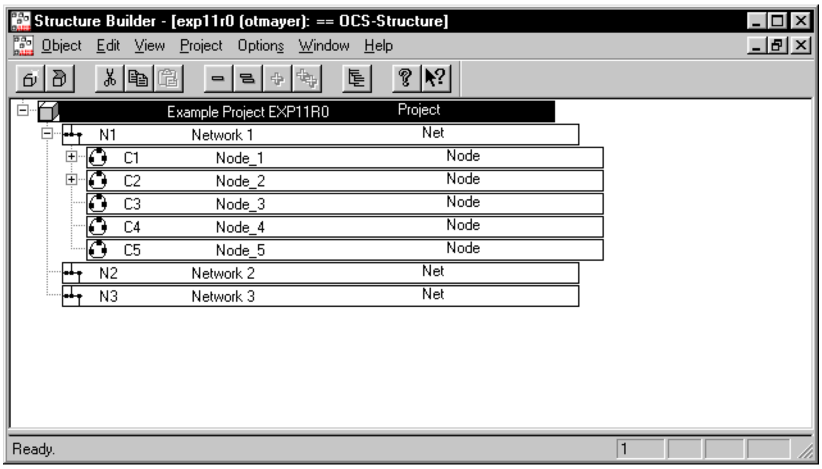
Based on the engineering data from where you start the project, you can estimate the main objects of the control system. Our example, the "spray cooling system", will be realized in a control system covering the control functions for the whole plant. Our example will only partially occupy a controller node as, for example, a part of an Advant Controller 450.

In a first engineering step, the type of the control equipment is defined and the number of control nodes estimated.

Prepare the basic structure for the control system. The top object is again the Project object. The next level is the network to which the controller is connected that controls the spray cooling system. Put this controller below the network. The PC programs are assigned to Node Control software aspects of controller objects. Children objects of the controller are objects that represent stations, boards and channels.

Place the boards, with their channels, below the node. The channels are presented as Advant objects. In future versions they may be presented as objects, that can be merged with signal objects.

Figure S4-2 shows an example of the basic structure of the control system.



	Example Project EXP11R0	Project
N1	Network 1	Net
C1	Node_1	Node
C2	Node_2	Node
C3	Node_3	Node
C4	Node_4	Node
C5	Node_5	Node
N2	Network 2	Net
N3	Network 3	Net

Figure S4-2. Basic Structure of a Control System

See in the relevant sections of Step 3 how to create objects in a structure and how to assign engineering data to objects.

S4.4 Setting the OCS Defaults

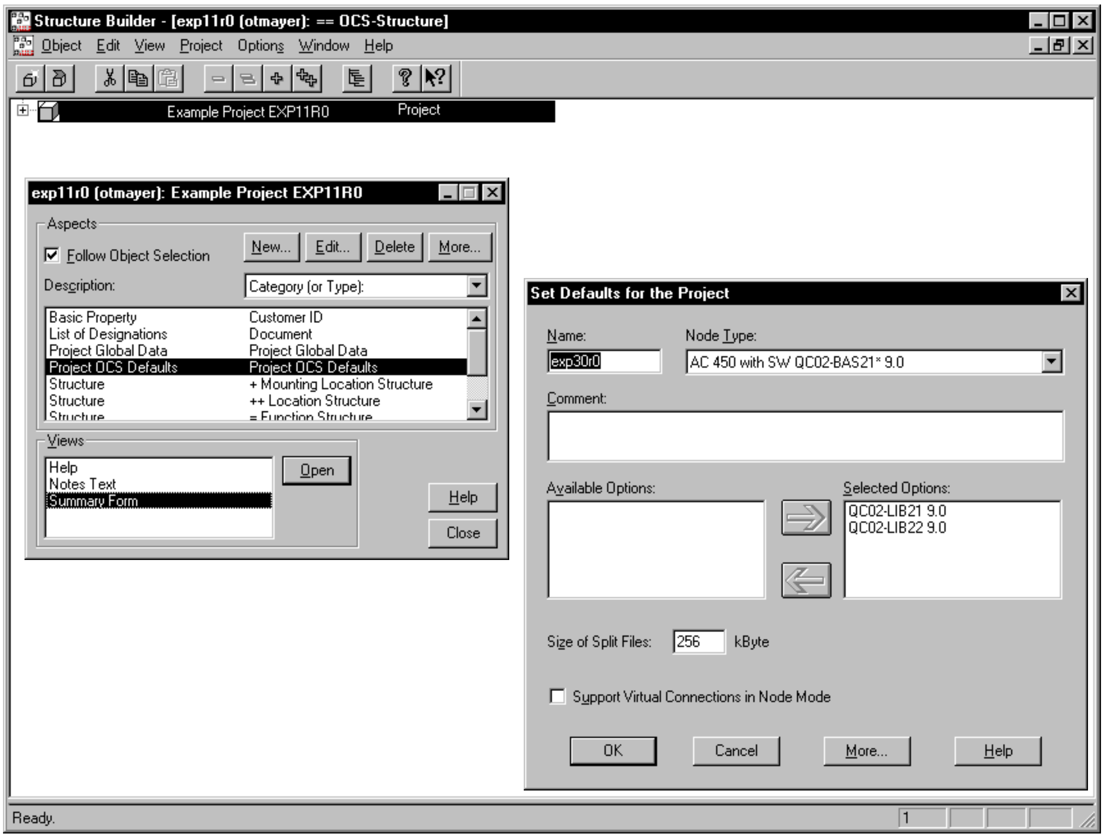
In the following we explain the setting of the OCS Defaults in the corresponding aspect of the Project object. This aspect is automatically created with the project object.

The OCS Defaults can be edited by opening the Summary Form view of this aspect. The Project Name, the default Node type and the default options are set in the **Set Defaults for the Project** window. The defaults can be set when the decision has been made about the most commonly used Node type in the project.

Setting the OCS-Defaults	
Preconditions:	Project object created

Setting the OCS-Defaults	
User activities:	1. Use the Structure Builder 2. Open the Object window of the Project object 3. Open the **Summary Form** view of the **Project OCS Defaults** aspect 4. Set the OCS Defaults in the **Set Defaults for the Project** dialog box 5. Click on **OK**
Result:	OCS Defaults are set and assigned to aspects representing OCS entities.

Figure S4-3 illustrates the procedure of setting OCS Defaults.


Structure Builder - [exp11r0 (otmayer): == OCS-Structure]
Object Edit View Project Options Window Help
Example Project EXP11R0
Project
exp11r0 (otmayer): Example Project EXP11R0
Aspects
New...
Edit...
Delete
More...
Views
Help
Notes Text
Summary Form
Open
Help
Close
Set Defaults for the Project
Available Options:
OK
Cancel
More...
Help
Ready.
1

☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
List of Designations	Document
Project Global Data	Project Global Data
Project OCS Defaults	Project OCS Defaults
Structure	+ Mounting Location Structure
Structure	++ Location Structure
Structure	= Function Structure

Name: exp30r0

Node Type: AC 450 with SW QC02-BAS21* 9.0

Comment:

Selected Options:
QC02-LIB21 9.0
QC02-LIB22 9.0

Size of Split Files: 256 kByte

☐ Support Virtual Connections in Node Mode

Figure S4-3. Setting the OCS Defaults

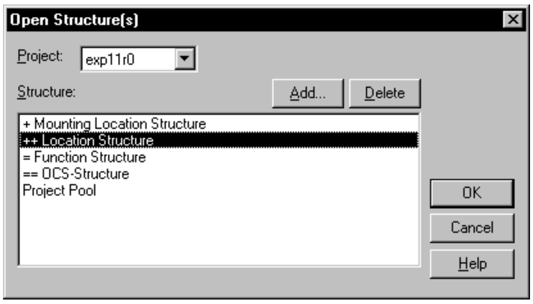
Step 5 Build the Location Structure

5.1 Opening the Location Structure

Start building up the Location Oriented Structure. Open the window of the Location Oriented Structure.

Opening a Structure Window	
Preconditions:	Project open
User activities:	1. Use the Structure Builder 2. Choose **Open Structure** in the **Window** menu 3. Select **Location Structure** in the **Open Structure** window 4. Click on **Open**
Result:	Location Structure window opens

Figure S5-1 shows the Open Structure window from where the structure to be opened is selected. You can select more than one structure.


Open Structure(s)

Project: exp11r0

Structure:

+ Mounting Location Structure
++ Location Structure
= Function Structure
== OCS-Structure
Project Pool

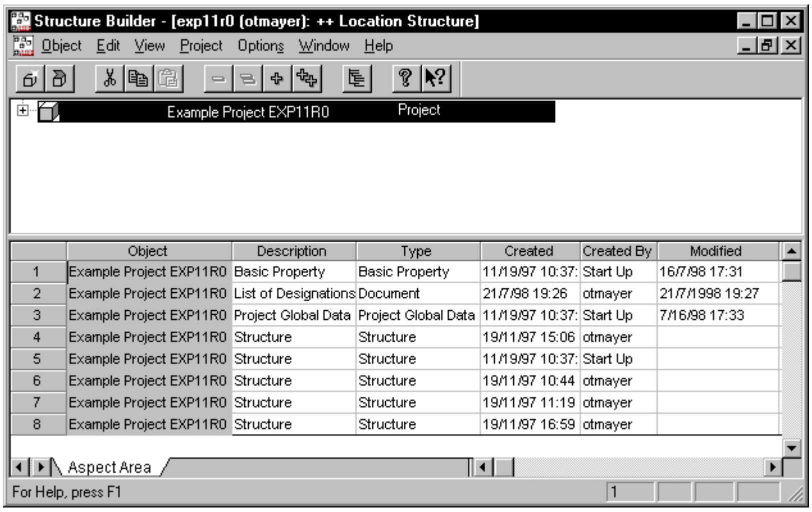
Figure S5-1. Opening the Location Structure in the Open Structure Window

S5.2 Placing the Project Object in the Location Structure

The Location Structure window is still empty. The top object in this structure shall be the project object. This object has already been created and is located in the project pool. Place this object into the Location Oriented Structure.

Placing the Project Object as Top Object of a Structure	
Preconditions:	Project created and opened
User activities:	1. Click on the **Project** object in the **Project Pool** window 2. Drag the **Project** object by pressing the right mouse button and drop it into the **Location Structure** window.
Result:	Project object is root object in the Location Structure

Figure S5-2 shows the Location Structure window with the placed project object.



Structure Builder - [exp11r0 (otmayer): ++ Location Structure]
Object Edit View Project Options Window Help
Example Project EXP11R0
Project
Aspect Area
For Help, press F1
1

	Object	Description	Type	Created	Created By	Modified
1	Example Project EXP11R0	Basic Property	Basic Property	11/19/97 10:37:	Start Up	16/7/98 17:31
2	Example Project EXP11R0	List of Designations	Document	21/7/98 19:26	otmayer	21/7/1998 19:27
3	Example Project EXP11R0	Project Global Data	Project Global Data	11/19/97 10:37:	Start Up	7/16/98 17:33
4	Example Project EXP11R0	Structure	Structure	19/11/97 15:06	otmayer	
5	Example Project EXP11R0	Structure	Structure	11/19/97 10:37:	Start Up	
6	Example Project EXP11R0	Structure	Structure	19/11/97 10:44	otmayer	
7	Example Project EXP11R0	Structure	Structure	19/11/97 11:19	otmayer	
8	Example Project EXP11R0	Structure	Structure	19/11/97 16:59	otmayer	

Figure S5-2. Location Structure Window with Placed Project Object

S5.3 Creation of Objects in the Location Structure

The top level of this structure is again the Advant object “project” representing in this structure the whole plant area.

On the following level, arrange the main areas of the plant:

- Casting machine area
- Ladle turret area
- Switch house area
- Adjusting area
- Rollout area
- Water distribution area.

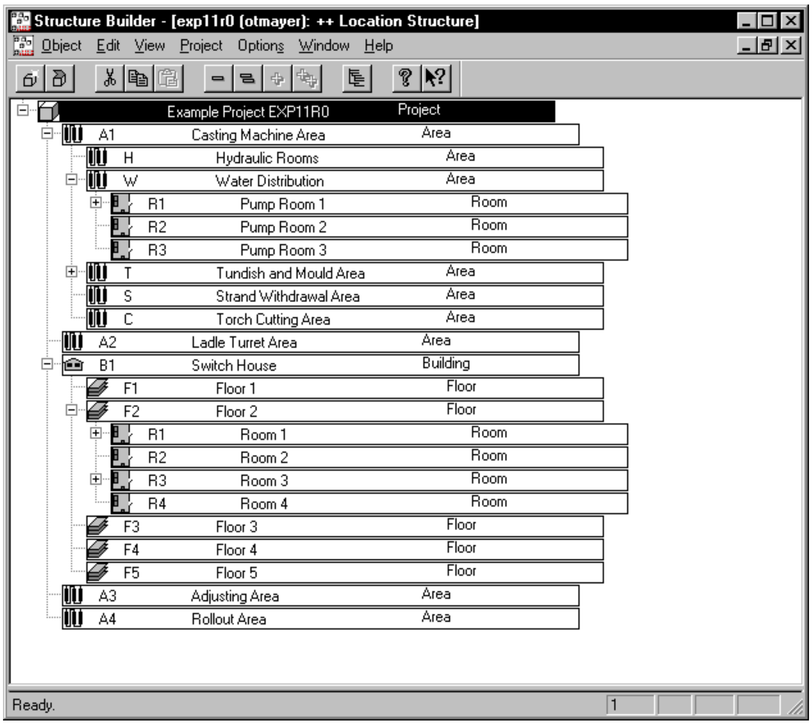
Create these objects directly into the Location Oriented Structure

In our example, we structure further down the locations related to the casting machine area. These locations are

- Water distribution area
- Hydraulic rooms
- Tundish and mould area
- Strand withdrawal area
- Torch cutting area.

We also structure further down the switch house into its floors and the floors into its rooms.

Figure S5-3 shows the corresponding Location Oriented Structure.


Structure Builder - [exp11r0 (otmayer): ++ Location Structure]
Object
Edit
View
Project
Options
Window
Help
Ready.
1

```
Example Project EXP11R0              Project
  A1  Casting Machine Area           Area
    H   Hydraulic Rooms              Area
    W   Water Distribution           Area
      R1  Pump Room 1                Room
      R2  Pump Room 2                Room
      R3  Pump Room 3                Room
    T   Tundish and Mould Area       Area
    S   Strand Withdrawal Area       Area
    C   Torch Cutting Area           Area
  A2  Ladle Turret Area              Area
  B1  Switch House                   Building
    F1  Floor 1                      Floor
    F2  Floor 2                      Floor
      R1  Room 1                     Room
      R2  Room 2                     Room
      R3  Room 3                     Room
      R4  Room 4                     Room
    F3  Floor 3                      Floor
    F4  Floor 4                      Floor
    F5  Floor 5                      Floor
  A3  Adjusting Area                 Area
  A4  Rollout Area                   Area
```

Figure S5-3. Location Structure for the Example Project

S5.4 Assigning Engineering Data to Location Objects

As mentioned, all the information regarding arrangements of devices, components and constructional units and its interconnection is related to locations. A typical document that covers the basic information for arrangement and interconnection is the Overview Diagram *Arrangement and Interconnection*. This document is created in the early phase of the engineering.

Figure S5-4 shows an example of such a document.

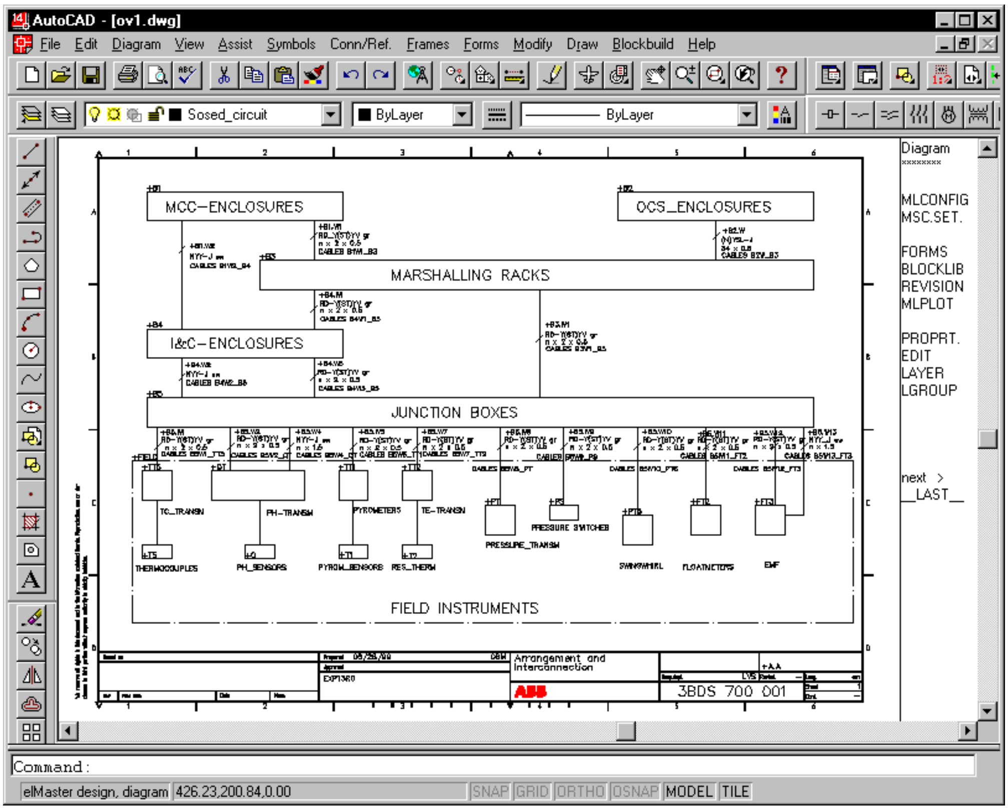


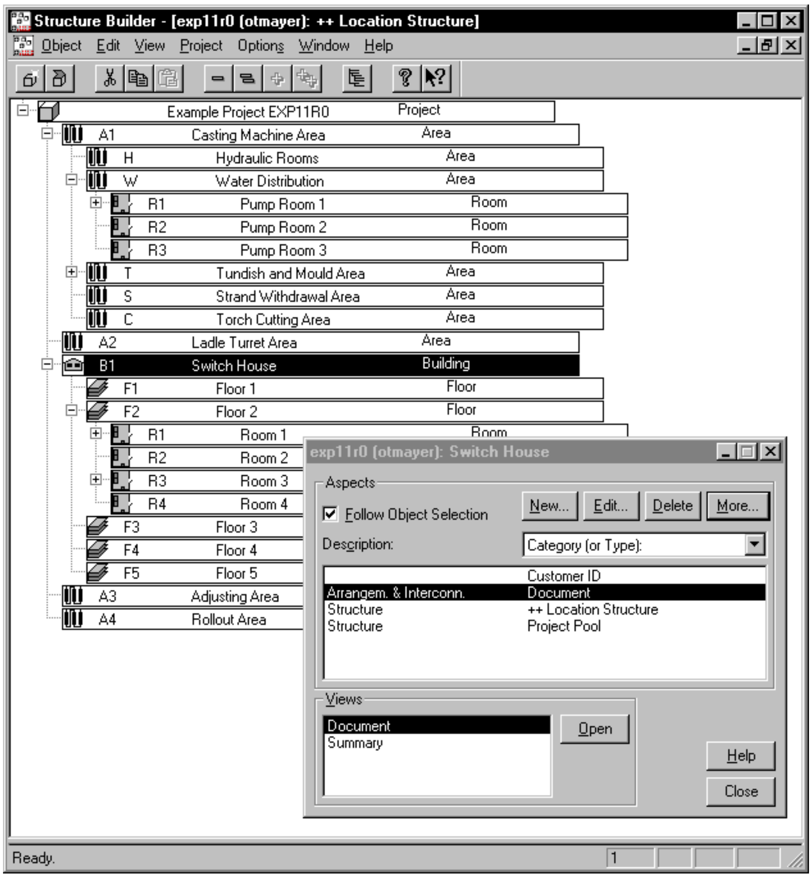
Figure S5-4. Overview of Arrangement and Interconnection

Before you start creating this document you have to decide to which Advant object the data, presented by this document, belong. This document shows the main locations for electrical and control equipment and their cable interconnections. As it covers either the locations of the whole plant, or of a subplant, it can be assigned either to the Advant object plant or to an Advant object representing the relevant subplant.

You can also assign the diagram to a location which has an outstanding importance with regard to arrangement and interconnection. Such an outstanding Advant object in our example is the switch house where all central parts of the electrical and control equipment are located. You can therefore assign the overview arrangement and interconnection data to the Advant object "switch house" (but it could also be assigned to the project object). Assigning data to an Advant object means to put this data into an aspect of this Advant object.

In our example, we create, in the Advant object Switch House, an aspect of type Document and insert as aspect description "Arrangement & Interconnection".

Figure S5-5 shows the Topographic Location Structure with the open Object window of the Advant object Switch House. The basic data of arrangement and interconnection are assigned as Document aspect to this object.


Structure Builder - [exp11r0 (otmayer): ++ Location Structure]
Object Edit View Project Options Window Help
Ready.
1
exp11r0 (otmayer): Switch House
Aspects
New...
Edit...
Delete
More...
Views
Open
Help
Close

Example Project EXP11R0 Project
A1 Casting Machine Area Area
H Hydraulic Rooms Area
W Water Distribution Area
R1 Pump Room 1 Room
R2 Pump Room 2 Room
R3 Pump Room 3 Room
T Tundish and Mould Area Area
S Strand Withdrawal Area Area
C Torch Cutting Area Area
A2 Ladle Turret Area Area
B1 Switch House Building
F1 Floor 1 Floor
F2 Floor 2 Floor
R1 Room 1 Room
R2 Room 2
R3 Room 3
R4 Room 4
F3 Floor 3
F4 Floor 4
F5 Floor 5
A3 Adjusting Area
A4 Rollout Area

☑ Follow Object Selection

Description: Category (or Type):

	Customer ID
Arrangem. & Interconn.	Document
Structure	++ Location Structure
Structure	Project Pool

Views: Document, Summary

Figure S5-5. Open Object Window with Aspect for Arrangement and Interconnection

The detailed location oriented engineering data is assigned to aspects of Advant objects representing the locations of, for example, interconnection terminals, devices, components and constructional units. This is done in the final design phase.

See more detailed information on how to create objects and how to assign engineering data to projects in the relevant sections of Step 3.

Step 6 Build New / Adapt Standard Solutions

S6.1 Working with Standard Solutions

We have created the basic part of the Function Oriented Structure and we have assigned the basic specification data to these objects. Now we can start realizing the specified functions.

The normal way to continue the work is to select preconfigured standards out of an Advant object library that was created project specific. A project specific Advant object library is a subset of a general reference library and contains the standard solutions and components that can be used in a specific project.

We will use standard solutions for:

- Flow control loop
- Pressure monitoring system
- Drive system for a pump.

Assume that you have to create the content of the Advant object library needed for our example. Create reusable solutions and put them into the object library. Instantiate the solutions by importing or copying them from the object library into the project structures and by adding instance specific data. The basic structure of the object library we use was imported as described in Step 2.

A standard solution consists of a set of Advant objects with its aspects. This set of Advant objects is placed as a substructure in the object library. The instantiating procedure consists of copying the set of Advant objects within its structure, adding of instance specific aspect data, and placing it in the relevant target structure. Use the Import functionality of the Structure Builder to perform this.

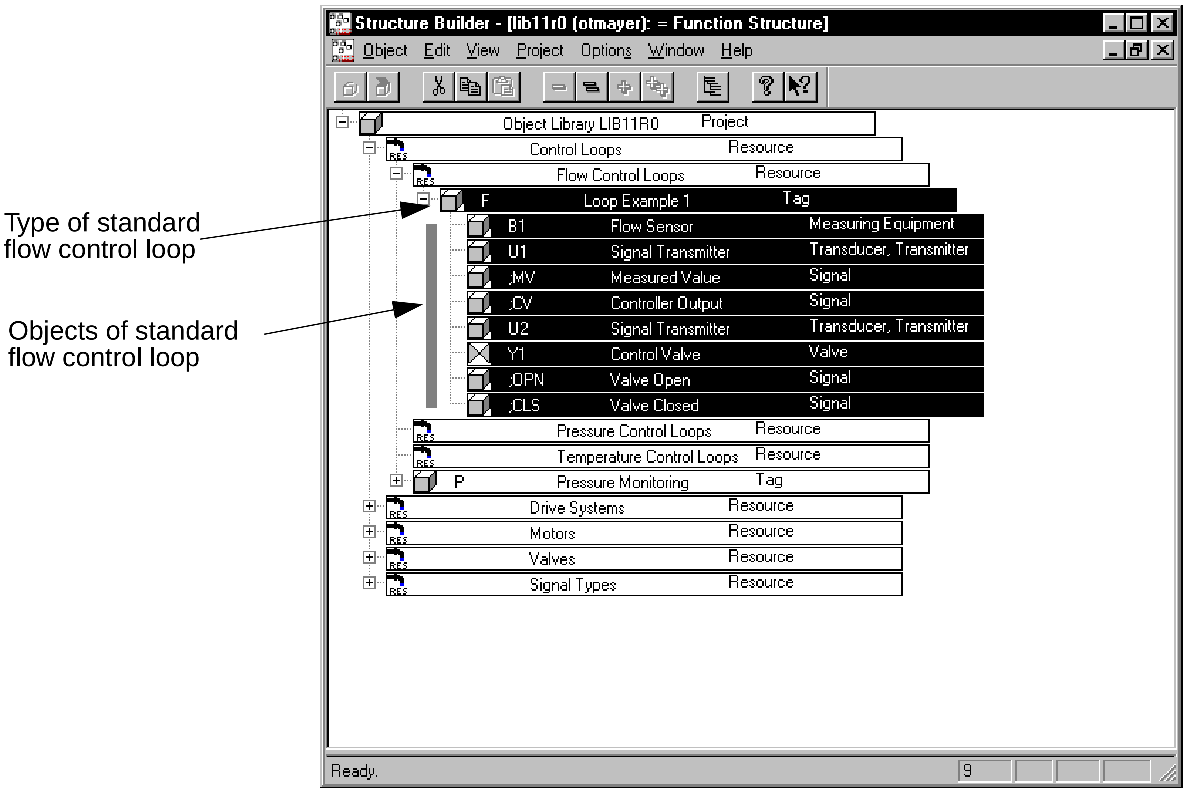
The object library can be created and administered as a project by its own. (See Step 1 Build Project Environment).

S6.2 Creating a Standard Control Loop

Configure a standard control loop for cooling water. The standard is based on a measuring principle, a standard signal level, a certain type of controller function and a certain type of actuating device. Furthermore it is based on a standard arrangement and interconnection concept.

S6.2.1 Advant Objects of the Standard Control Loop in the Object Library

Figure S6-1 shows the standard flow control loop in the Function Structure of the object library.


Structure Builder - [lib11r0 (otmayer): = Function Structure]
Object
Edit
View
Project
Options
Window
Help
Type of standard flow control loop
Objects of standard flow control loop
Ready.
9

	Object Library LIB11R0	Project
	Control Loops	Resource
	Flow Control Loops	Resource
F	Loop Example 1	Tag
B1	Flow Sensor	Measuring Equipment
U1	Signal Transmitter	Transducer, Transmitter
;MV	Measured Value	Signal
;CV	Controller Output	Signal
U2	Signal Transmitter	Transducer, Transmitter
Y1	Control Valve	Valve
;OPN	Valve Open	Signal
;CLS	Valve Closed	Signal
	Pressure Control Loops	Resource
	Temperature Control Loops	Resource
P	Pressure Monitoring	Tag
	Drive Systems	Resource
	Motors	Resource
	Valves	Resource
	Signal Types	Resource

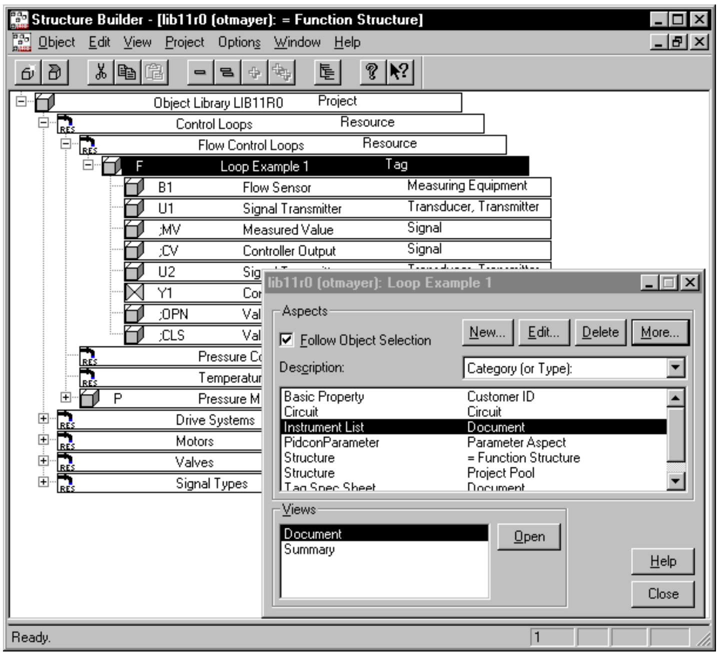
Figure S6-1. Standard Flow Control Loop in the Object Library

After creating the objects, add the aspects and predefine the aspect attributes. The aspect data that are not instance specific can be filled in. The instance specific data is filled in at the instances, which are created, for example, by copying the control loop subtree in the object library.

S6.2.2 Examples of Advant Object Aspects in the Standard Control Loop

S6.2.2.1 Document Aspect

Figure S6-2 shows an Advant object in the Function Structure representing a standard tag of the object library. Its object window shows a document aspect described as instrument list. The aspect data is presented in an Excel spreadsheet, Figure S6-3. After clicking on Document Open the document is presented for inspection or editing. Some of the fields for aspect data are empty because these data are instance specific and filled in during the detail engineering phase.



	Object Library LIB11R0	Project
	Control Loops	Resource
	Flow Control Loops	Resource
F	Loop Example 1	Tag
B1	Flow Sensor	Measuring Equipment
U1	Signal Transmitter	Transducer, Transmitter
;MV	Measured Value	Signal
;CV	Controller Output	Signal
U2	Sig	
Y1	Cor	
;OPN	Val	
;CLS	Val	
	Pressure Co	
	Temperatur	
P	Pressure M	
	Drive Systems	
	Motors	
	Valves	
	Signal Types	

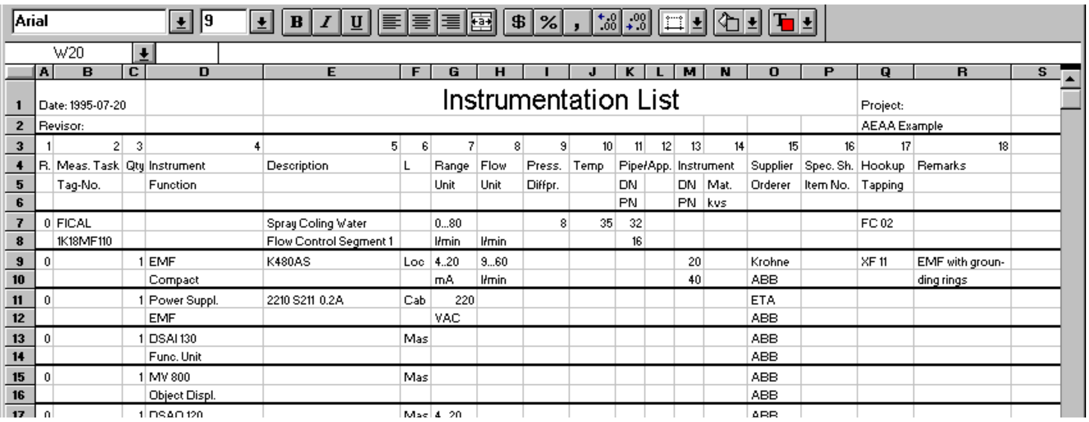
☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
Circuit	Circuit
Instrument List	Document
PidconParameter	Parameter Aspect
Structure	= Function Structure
Structure	Project Pool
Tag Spec Sheet	Document

Views: Document, Summary

Figure S6-2. Selecting a Document View for Editing



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Date: 1995-07-20				Instrumentation List												Project:		
2	Revisor:																AEAA Example		
3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
4	R.	Meas. Task	Qty	Instrument	Description	L	Range	Flow	Press.	Temp	Pipe/App.		Instrument		Supplier	Spec. Sh.	Hookup	Remarks	
5		Tag-No.		Function			Unit	Unit	Diffpr.		DN		DN	Mat.	Orderer	Item No.	Tapping		
6											PN		PN	kvs					
7	0	FICAL			Spray Coling Water		0...80		8	35	32						FC 02		
8		1K18MF110			Flow Control Segment 1		l/min	l/min			16								
9	0		1	EMF	K480AS	Loc	4..20	9...60					20		Krohne		XF 11	EMF with groun-	
10				Compact			mA	l/min					40		ABB			ding rings	
11	0		1	Power Suppl.	2210 S211 0.2A	Cab	220								ETA				
12				EMF			VAC								ABB				
13	0		1	DSAI 130		Mas									ABB				
14				Func. Unit											ABB				
15	0		1	MV 800		Mas									ABB				
16				Object Displ.											ABB				
17	0		1	DSAO 120		Mas	4..20								ABB				

Figure S6-3. Example for the Document View of a Document Aspect

S6.2.2.2 Circuit Aspect

Assign the control algorithm to the object that represents the standard control loop. It shall be realized in an AC 450 controller of the OCS-system. Add a circuit aspect to the tag object. It can later be allocated to a PC-program in the Node Control Software aspect of an AC 450-node.

Creation of a Circuit Aspect and Its Control logic	
Preconditions:	Advant object created to which circuit aspect can be added
User activities:	1. Use the Structure Builder 2. Click on the object to which the circuit aspect shall be added. 3. Open the **Object** window and click on **New** 4. Select **Circuit** in the **New Aspect** window and fill in the **Aspect Description** 5. Click on **OK** 6. Click on (the new inserted) **Circuit** in the Aspect Window 7. Open **Function Chart** view 8. Design control logic for the circuit with the Function Chart Builder and generate Source Code 9. Exit the Function Chart Builder
Result:	Circuit Aspect and its control logic created

The circuit may contain a Type Circuit in which symbols were defined. The symbols will be substituted by instance specific parameters at the instantiation of the circuit. The parameters are defined in a separate parameter aspect that is placed in the same object as the circuit.

By the same principles, create the standard solution for the pressure monitoring system.

Figure S6-4 shows the Function Structure window of the object library, the Object window and the New Aspect window as presented when adding the Circuit aspect. The circuit of this example contains a type circuit.

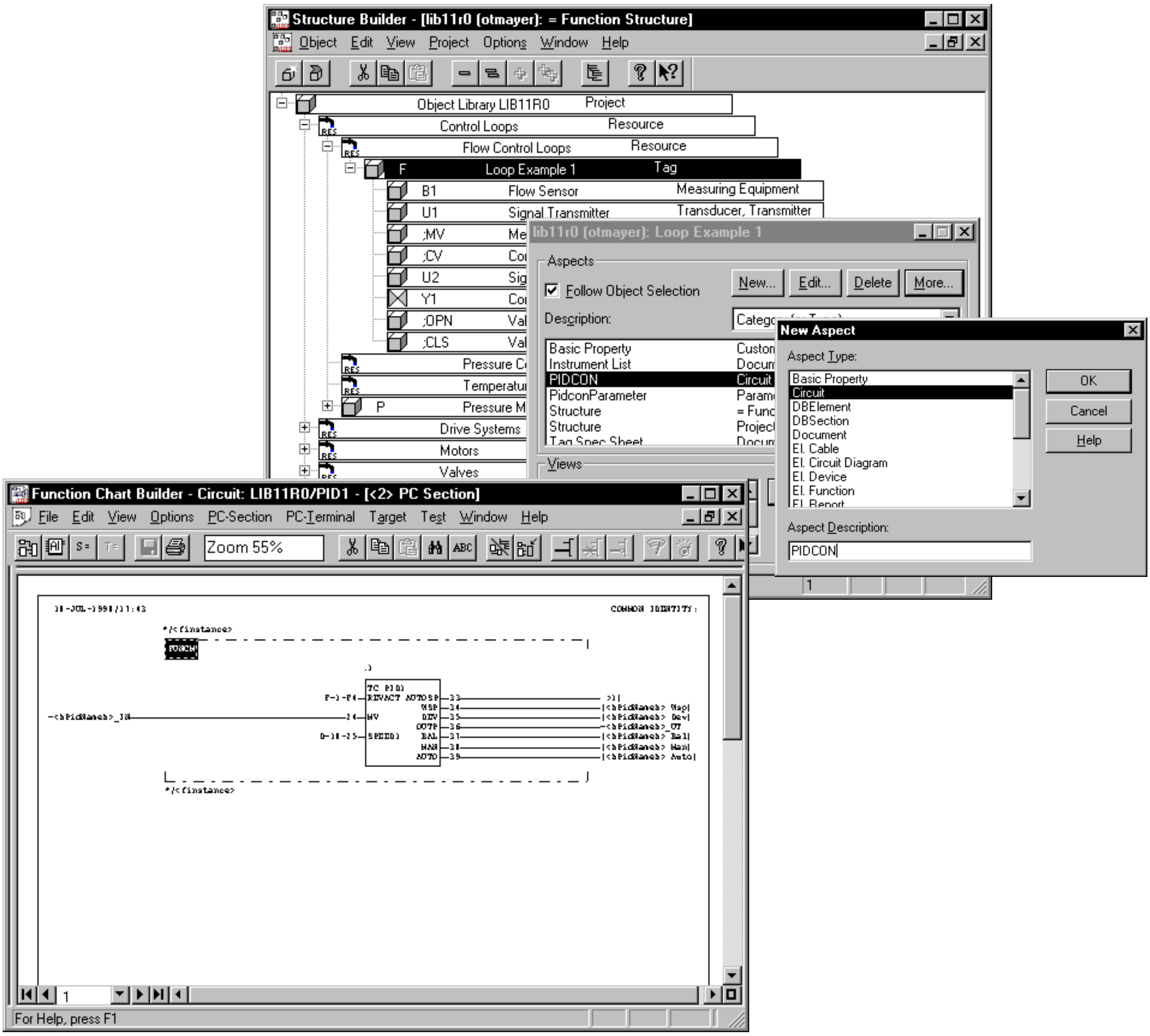

Structure Builder - [lib11r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
Object Library LIB11R0 Project
Control Loops Resource
Flow Control Loops Resource
F Loop Example 1 Tag
B1 Flow Sensor Measuring Equipment
U1 Signal Transmitter Transducer, Transmitter
;MV
;CV
U2
Y1
;OPN
;CLS
Pressure C
Temperatu
P Pressure M
Drive Systems
Motors
Valves
lib11r0 (otmayer): Loop Example 1
Aspects
☑ Follow Object Selection
New... Edit... Delete More...
Description:
Basic Property
Instrument List
PIDCON Circuit
PidconParameter
Structure
Structure
Tag Spec Sheet
Views
New Aspect
Aspect Type:
Basic Property
Circuit
DBElement
DBSection
Document
El. Cable
El. Circuit Diagram
El. Device
El. Function
El. Report
OK
Cancel
Help
Aspect Description:
PIDCON
Function Chart Builder - Circuit: LIB11R0/PID1 - [<2> PC Section]
File Edit View Options PC-Section PC-Terminal Target Test Window Help
Zoom 55%
PIDCON
For Help, press F1

Figure S6-4. Adding a Circuit Aspect to an Advant Object

S6.2.2.3 Electrical Circuit Diagram Aspect

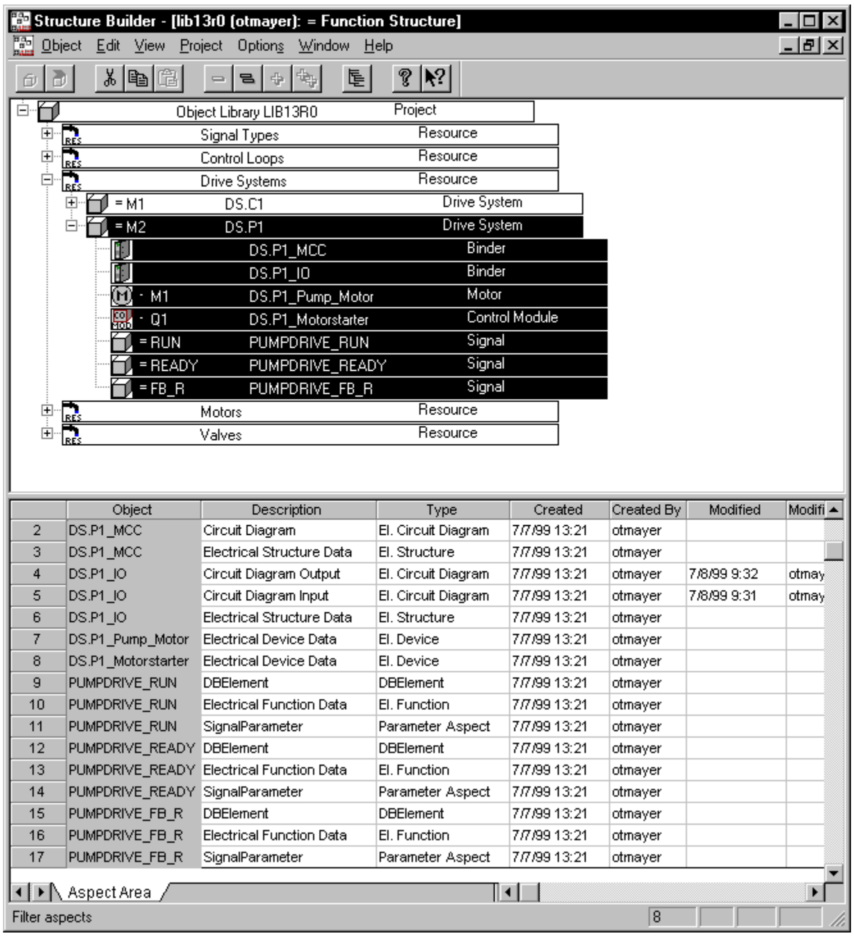
This aspect is not considered in this example to keep the focus on the OCS part. A real library solution would contain a typical for a Loop Diagram.

S6.3 Creating a Standard Drive System

We configure now a standard drive system that we can use for the pump drive in our spray cooling system.

S6.3.1 Advant Objects of the Standard Drive System in the Object Library

Figure S6-5 shows the standard drive system in the Function Structure of the object library. The aspect area of the Structure Builder window is presented in this figure. This area shows the aspects of the selected objects of the standard drive system. Structure aspects and Basic Property aspects of the Structure Builder are filtered out and not presented.


Structure Builder - [lib13r0 (otmayer): = Function Structure]
Object
Edit
View
Project
Options
Window
Help

	Object Library LIB13R0	Project
	Signal Types	Resource
	Control Loops	Resource
	Drive Systems	Resource
= M1	DS.C1	Drive System
= M2	DS.P1	Drive System
	DS.P1_MCC	Binder
	DS.P1_IO	Binder
- M1	DS.P1_Pump_Motor	Motor
- Q1	DS.P1_Motorstarter	Control Module
= RUN	PUMPDRIVE_RUN	Signal
= READY	PUMPDRIVE_READY	Signal
= FB_R	PUMPDRIVE_FB_R	Signal
	Motors	Resource
	Valves	Resource

	Object	Description	Type	Created	Created By	Modified	Modifi
2	DS.P1_MCC	Circuit Diagram	El. Circuit Diagram	7/7/99 13:21	otmayer		
3	DS.P1_MCC	Electrical Structure Data	El. Structure	7/7/99 13:21	otmayer		
4	DS.P1_IO	Circuit Diagram Output	El. Circuit Diagram	7/7/99 13:21	otmayer	7/8/99 9:32	otmay
5	DS.P1_IO	Circuit Diagram Input	El. Circuit Diagram	7/7/99 13:21	otmayer	7/8/99 9:31	otmay
6	DS.P1_IO	Electrical Structure Data	El. Structure	7/7/99 13:21	otmayer		
7	DS.P1_Pump_Motor	Electrical Device Data	El. Device	7/7/99 13:21	otmayer		
8	DS.P1_Motorstarter	Electrical Device Data	El. Device	7/7/99 13:21	otmayer		
9	PUMPDRIVE_RUN	DBElement	DBElement	7/7/99 13:21	otmayer		
10	PUMPDRIVE_RUN	Electrical Function Data	El. Function	7/7/99 13:21	otmayer		
11	PUMPDRIVE_RUN	SignalParameter	Parameter Aspect	7/7/99 13:21	otmayer		
12	PUMPDRIVE_READY	DBElement	DBElement	7/7/99 13:21	otmayer		
13	PUMPDRIVE_READY	Electrical Function Data	El. Function	7/7/99 13:21	otmayer		
14	PUMPDRIVE_READY	SignalParameter	Parameter Aspect	7/7/99 13:21	otmayer		
15	PUMPDRIVE_FB_R	DBElement	DBElement	7/7/99 13:21	otmayer		
16	PUMPDRIVE_FB_R	Electrical Function Data	El. Function	7/7/99 13:21	otmayer		
17	PUMPDRIVE_FB_R	SignalParameter	Parameter Aspect	7/7/99 13:21	otmayer		

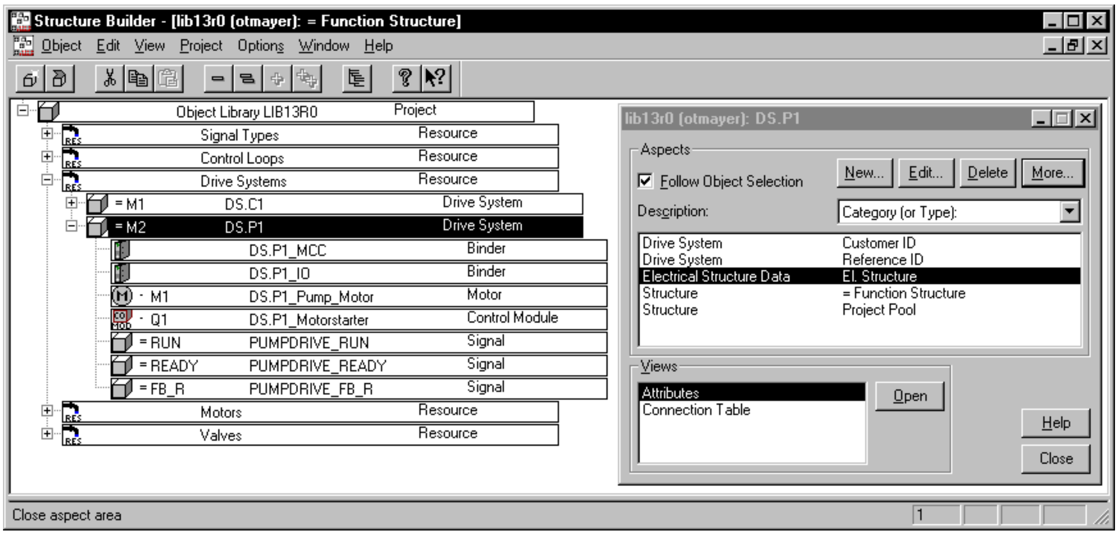
Figure S6-5. Standard Drive System in the Object Library

Object DS.P1 of Type Drive System

This object represents the entire drive system. All the information that concerns the drive system as a whole is assigned to aspects of this object. Such information are for example

- Descriptions of the drive system
- Control Logic of the drive system
- Designation of the drive system
- Structure information about the drive system.

As it is a standard that shall be used in applications, only the typical information is given. The application specific information is inserted when the standard solution is used. Figure S6-6 shows the position of this object in the Function Structure and its aspects in the object window.



	Object Library LIB13R0	Project
	Signal Types	Resource
	Control Loops	Resource
	Drive Systems	Resource
= M1	DS.C1	Drive System
= M2	DS.P1	Drive System
	DS.P1_MCC	Binder
	DS.P1_IO	Binder
- M1	DS.P1_Pump_Motor	Motor
- Q1	DS.P1_Motorstarter	Control Module
= RUN	PUMPDRIVE_RUN	Signal
= READY	PUMPDRIVE_READY	Signal
= FB_R	PUMPDRIVE_FB_R	Signal
	Motors	Resource
	Valves	Resource

☑ Follow Object Selection

Description:	Category (or Type):
Drive System	Customer ID
Drive System	Reference ID
Electrical Structure Data	El. Structure
Structure	= Function Structure
Structure	Project Pool

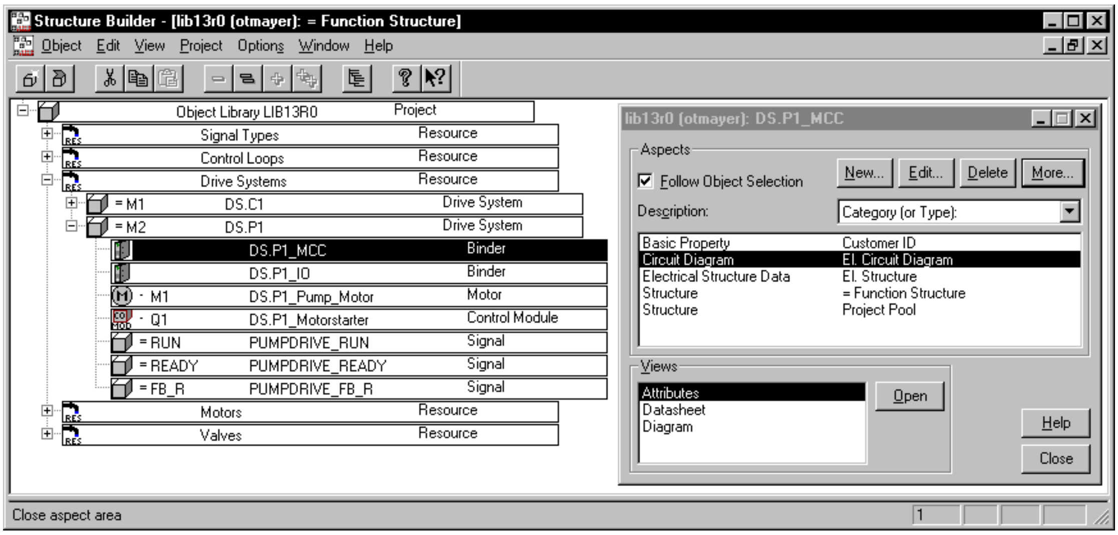
Views: Attributes, Connection Table

Figure S6-6. Object DS.P1 with Its Aspects

Object DS.P1_MCC of Type Binder

This object contains the Circuit Diagram of the mains supply of the motor. The Circuit Diagram is assigned to the aspect type El. Circuit Diagram. The Binder object shall be placed later on in the Mounting Location Structure to get the complete reference designations of the objects in the diagrams. This is the reason for not assigning this Circuit Diagram directly to the object DS.P1 Drive System. See also Step 19.

Figure S6-7 shows the position of this Binder object in the Function Structure and its aspects in the object window.



	Object Library LIB13R0	Project
	Signal Types	Resource
	Control Loops	Resource
	Drive Systems	Resource
= M1	DS.C1	Drive System
= M2	DS.P1	Drive System
	DS.P1_MCC	Binder
	DS.P1_IO	Binder
- M1	DS.P1_Pump_Motor	Motor
- Q1	DS.P1_Motorstarter	Control Module
= RUN	PUMPDRIVE_RUN	Signal
= READY	PUMPDRIVE_READY	Signal
= FB_R	PUMPDRIVE_FB_R	Signal
	Motors	Resource
	Valves	Resource

☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
Circuit Diagram	El. Circuit Diagram
Electrical Structure Data	El. Structure
Structure	= Function Structure
Structure	Project Pool

Views: Attributes, Datasheet, Diagram

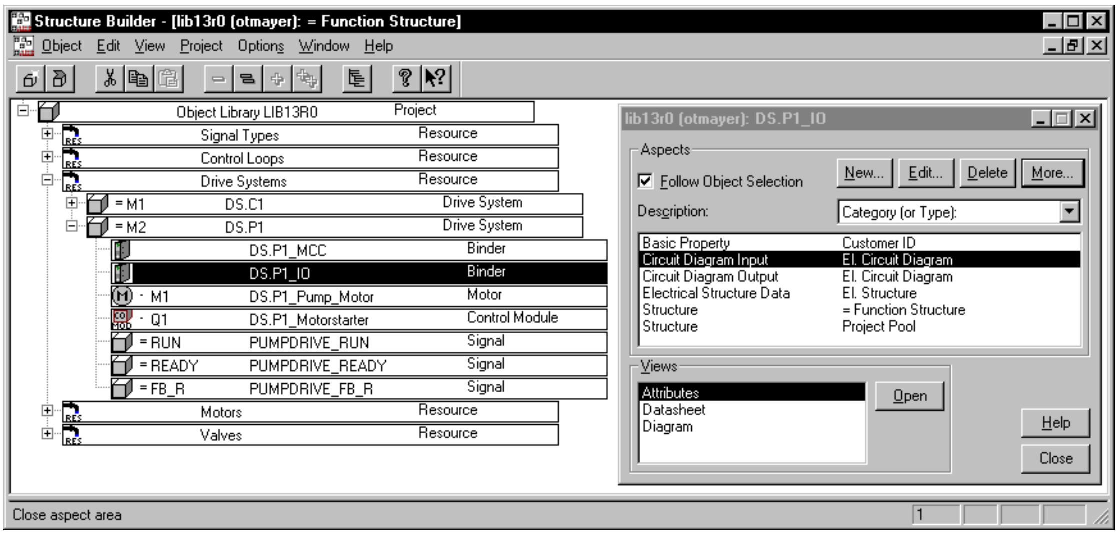
Figure S6-7. Object DS.C1_MCC Binder with Its Aspects

The following aspects are assigned to this object:

- Basic Property aspect Customer ID carrying the object name and object type
- El. Circuit Diagram aspect carrying the Circuit Diagram for the main motor circuit and the control circuits of the motorstarter
- Electrical Structure aspect for the exchange of structure information between the Structure Builder database and the El. Diagram Builder database
- Structure aspect carrying the reference designation and the information about the position in the Function Structure
- Structure aspect carrying the information that the object is placed in the project pool.

Object DS.P1_IO of Type Binder

This object contains the Circuit Diagram of the signals connecting the motorstarter and the OCS system. Separate Circuit Diagrams are used for input and output signals. The Circuit Diagrams are assigned to the aspect types El. Circuit Diagram. The Binder objects shall be placed later on in the Mounting Location Structure representing the cabinet that contains the relevant I/O boards. Thus the objects in the diagrams will get the complete reference designations.
See also Step 19.



	Object Library LIB13R0	Project
	Signal Types	Resource
	Control Loops	Resource
	Drive Systems	Resource
= M1	DS.C1	Drive System
= M2	DS.P1	Drive System
	DS.P1_MCC	Binder
	DS.P1_IO	Binder
- M1	DS.P1_Pump_Motor	Motor
- Q1	DS.P1_Motorstarter	Control Module
= RUN	PUMPDRIVE_RUN	Signal
= READY	PUMPDRIVE_READY	Signal
= FB_R	PUMPDRIVE_FB_R	Signal
	Motors	Resource
	Valves	Resource

☑ Follow Object Selection

Description:	Category (or Type):
Basic Property	Customer ID
Circuit Diagram Input	El. Circuit Diagram
Circuit Diagram Output	El. Circuit Diagram
Electrical Structure Data	El. Structure
Structure	= Function Structure
Structure	Project Pool

Views: Attributes, Datasheet, Diagram

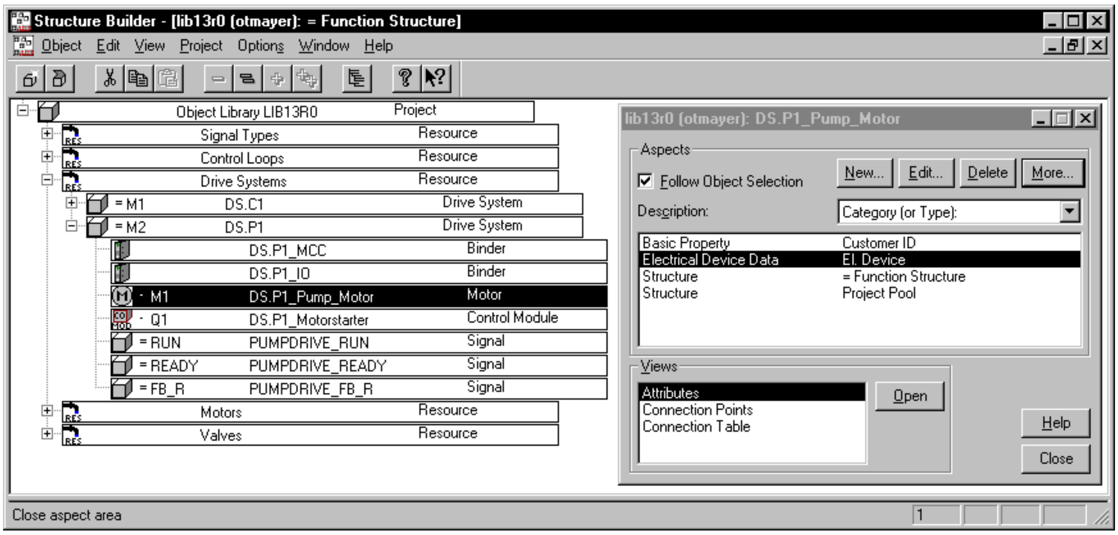
Figure S6-8. Object DS.P1_IO Binder with Its Aspects

The following aspects are assigned to this object:

- Basic Property aspect Customer ID carrying the object name and object type
- El. Circuit Diagram aspect carrying the Circuit Diagram for the input signals to the OCS-system
- El. Circuit Diagram aspect carrying the Circuit Diagram for the output signals to the OCS-system
- Electrical Structure aspect for the exchange of structure information between the Structure Builder database and the El. Diagram Builder database
- Structure aspect carrying the reference designation and the information about the position in the Function Structure
- Structure aspect carrying the information that the object is placed in the project pool.

Object DS.P1_Pump_Motor of Type Motor

This object contains the device data of a standard motor used for the standard drive system. The data are assigned by a reference from the aspect type El. Device to the corresponding article data.


Structure Builder - [lib13r0 (otmayer): = Function Structure]
Object
Edit
View
Project
Options
Window
Help
lib13r0 (otmayer): DS.P1_Pump_Motor
New...
Edit...
Delete
More...
Open
Help
Close

	Object Library LIB13R0	Project
	Signal Types	Resource
	Control Loops	Resource
	Drive Systems	Resource
= M1	DS.C1	Drive System
= M2	DS.P1	Drive System
	DS.P1_MCC	Binder
	DS.P1_IO	Binder
- M1	DS.P1_Pump_Motor	Motor
- Q1	DS.P1_Motorstarter	Control Module
= RUN	PUMPDRIVE_RUN	Signal
= READY	PUMPDRIVE_READY	Signal
= FB_R	PUMPDRIVE_FB_R	Signal
	Motors	Resource
	Valves	Resource

Aspects

☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
Electrical Device Data	El. Device
Structure	= Function Structure
Structure	Project Pool

Views

Attributes
Connection Points
Connection Table

Close aspect area

Figure S6-9. Object DS.P1_Main_Drive Motor with Its Aspects

The following aspects are assigned to this object:

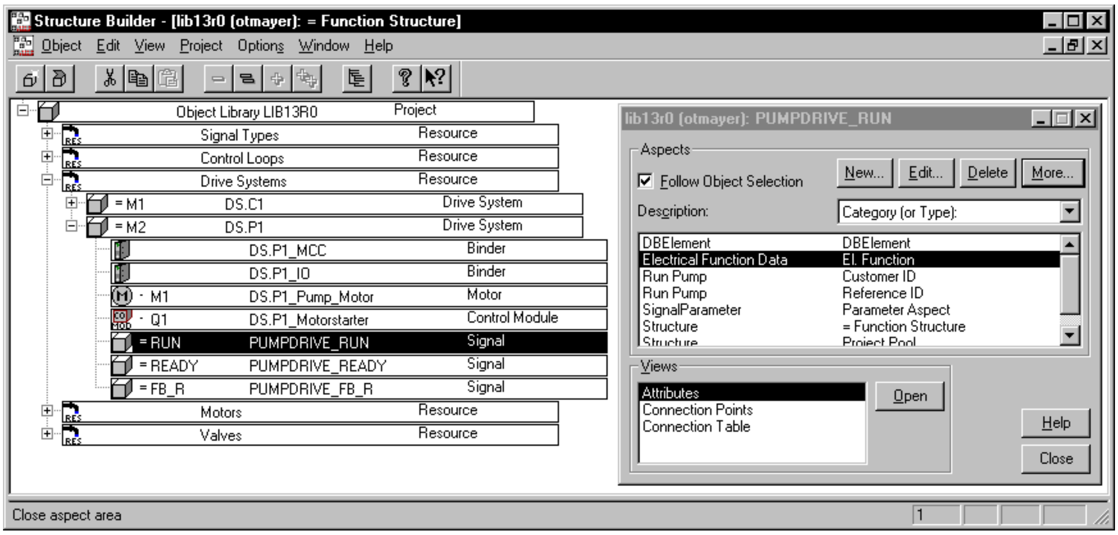
- Basic Property aspect Customer ID carrying the object name (which will be used as a typical motor name) and object type
- Electrical Device aspect to exchange data between database and symbols in diagrams and to assign article (product) data
- Structure aspect carrying the reference designation and the information about the position in the Function Structure
- Structure aspect carrying the information that the object is placed in the project pool.

Object DS.P1_Motorstarter of Type Control Module

This object carries the same aspect types as the object DS.P1_Pump_Motor. A motorstarter will be assigned to the Electrical Device aspect of this object.

Object DS.P1_RUN_R of Type Signal

This object contains the OCS and Circuit Diagram related information about the signal that gives the run command to the motor.


Structure Builder - [lib13r0 (otmayer): = Function Structure]
Object
Edit
View
Project
Options
Window
Help
lib13r0 (otmayer): PUMPDRIVE_RUN
Aspects
New...
Edit...
Delete
More...
Views
Open
Help
Close

	Object Library LIB13R0	Project
	Signal Types	Resource
	Control Loops	Resource
	Drive Systems	Resource
= M1	DS.C1	Drive System
= M2	DS.P1	Drive System
	DS.P1_MCC	Binder
	DS.P1_IO	Binder
- M1	DS.P1_Pump_Motor	Motor
- Q1	DS.P1_Motorstarter	Control Module
= RUN	PUMPDRIVE_RUN	Signal
= READY	PUMPDRIVE_READY	Signal
= FB_R	PUMPDRIVE_FB_R	Signal
	Motors	Resource
	Valves	Resource

☑ Follow Object Selection

Description: Category (or Type):

DBElement	DBElement
Electrical Function Data	El. Function
Run Pump	Customer ID
Run Pump	Reference ID
SignalParameter	Parameter Aspect
Structure	= Function Structure
Structure	Project Pool

Attributes
Connection Points
Connection Table

Close aspect area
1

Figure S6-10. Object DS.C1_RUN_R I/O-Signal with Its Aspects

The following aspects are assigned to this object:

- Basic Property aspect Customer ID carrying the object name (which will be used as signal name in the OCS system) and object type
- Basic Property aspect Reference ID carrying reference information to the source object when it is used by import in applications
- Database Element aspect that contains or references to a type circuit where the DB Element is defined
- Parameter aspect to define the signal parameters. The parameter values are not filled in. The defaults for the parameters are defined in the type circuit for the database element aspect. The application specific parameter values are set when the signal is used in an application.
- Electrical Function aspect to exchange data between database and symbols in diagrams
- Structure aspect carrying the reference designation and the information about the position in the Function Structure
- Structure aspect carrying the information that the object is placed in the project pool.

Object DS.P1_READY of Type Signal

Object DS.P1_FB_RUN of Type Signal

The above listed objects carry the same aspect types as the object DS.P1_RUN. Their contents represent the typical signal data which are adapted and completed when the drive system is used in an application.

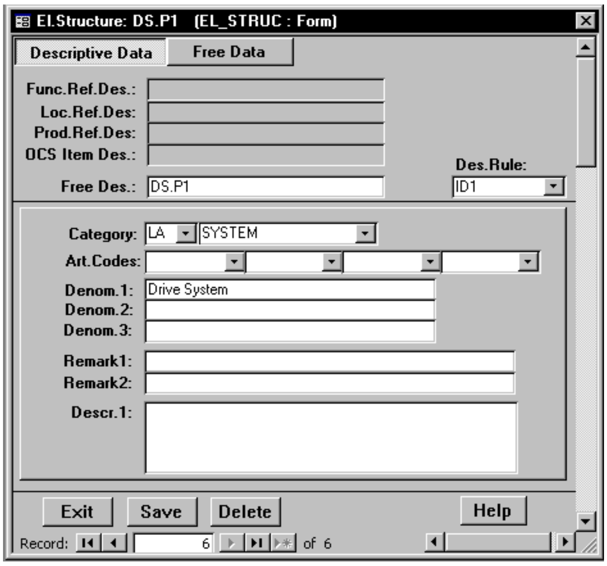
S6.3.2 Examples of Advant Object Aspects in the Standard Drive System

S6.3.2.1 Aspects Related to the Electrical Diagram Builder

Aspects for objects handled by the Electrical Diagram Builder are those types of aspects, which can be connected and linked to symbols in diagrams and therefore exchange data between the database and symbols in diagrams. In the following we call it electrical plant object aspects. These are aspects of the following types:

Electrical Structure Aspect

This aspect is set in objects of the library which will be used in applications where the Electrical Diagram Builder is used and no other electrical plant object aspect is inserted. Figure S6-11 shows the Form view of an Electrical Structure aspect. This aspect carries the structure information of electrical plant objects.



El.Structure: DS.P1 (EL_STRUC : Form)

Free Des.: DS.P1

Des.Rule: ID1

Category: LA SYSTEM

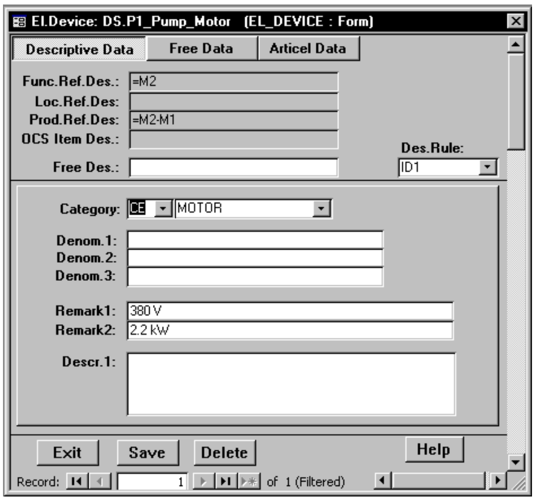
Denom.1: Drive System

Record: 6 of 6

Figure S6-11. Form View of Electrical Structure Aspect Data

Electrical Device Aspect

This aspect will be inserted in objects that represent an electrical device and to which articles shall be assigned. Figure S6-12 shows the Form view of an electrical device aspects in the Object Library example.



El.Device: DS.P1_Pump_Motor (EL_DEVICE : Form)

Func.Ref.Des.: =M2
Loc.Ref.Des:
Prod.Ref.Des: =M2-M1
OCS Item Des.:
Free Des.:
Des.Rule: ID1

Category: CE MOTOR

Denom.1:
Denom.2:
Denom.3:

Remark1: 380 V
Remark2: 2.2 kW

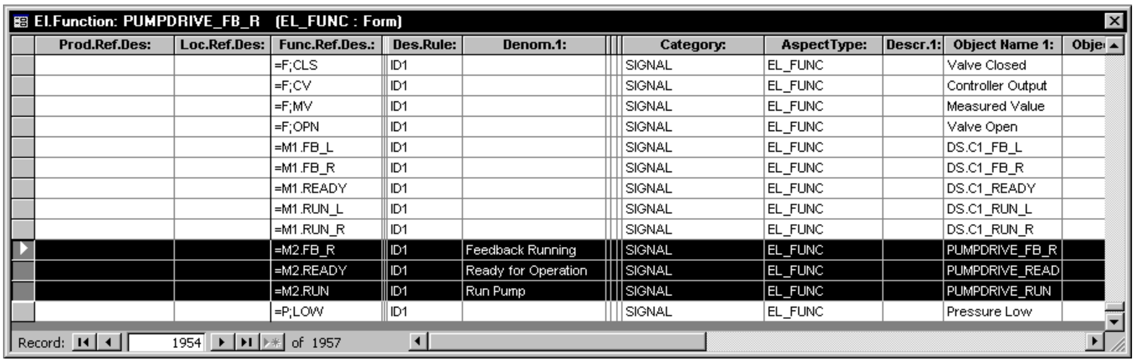
Descr.1:

Record: 1 of 1 (Filtered)

Figure S6-12. Form View of Electrical Device Aspects

Electrical Function Aspect

This aspect is inserted in objects that represent signals and channels that shall be handled by the El. Diagram Builder. Figure S6-13 shows the datasheet view of all electrical function aspects in the object library example.



El.Function: PUMPDRIVE_FB_R (EL_FUNC : Form)

Prod.Ref.Des:	Loc.Ref.Des:	Func.Ref.Des.:	Des.Rule:	Denom.1:	Category:	AspectType:	Descr.1:	Object Name 1:	Obje
		=F;CLS	ID1		SIGNAL	EL_FUNC		Valve Closed	
		=F;CV	ID1		SIGNAL	EL_FUNC		Controller Output	
		=F;MV	ID1		SIGNAL	EL_FUNC		Measured Value	
		=F;OPN	ID1		SIGNAL	EL_FUNC		Valve Open	
		=M1.FB_L	ID1		SIGNAL	EL_FUNC		DS.C1_FB_L	
		=M1.FB_R	ID1		SIGNAL	EL_FUNC		DS.C1_FB_R	
		=M1.READY	ID1		SIGNAL	EL_FUNC		DS.C1_READY	
		=M1.RUN_L	ID1		SIGNAL	EL_FUNC		DS.C1_RUN_L	
		=M1.RUN_R	ID1		SIGNAL	EL_FUNC		DS.C1_RUN_R	
		=M2.FB_R	ID1	Feedback Running	SIGNAL	EL_FUNC		PUMPDRIVE_FB_R	
		=M2.READY	ID1	Ready for Operation	SIGNAL	EL_FUNC		PUMPDRIVE_READ	
		=M2.RUN	ID1	Run Pump	SIGNAL	EL_FUNC		PUMPDRIVE_RUN	
		=P;LOW	ID1		SIGNAL	EL_FUNC		Pressure Low	

Record: 1954 of 1957

Figure S6-13. Datasheet View of Electrical Function Aspects

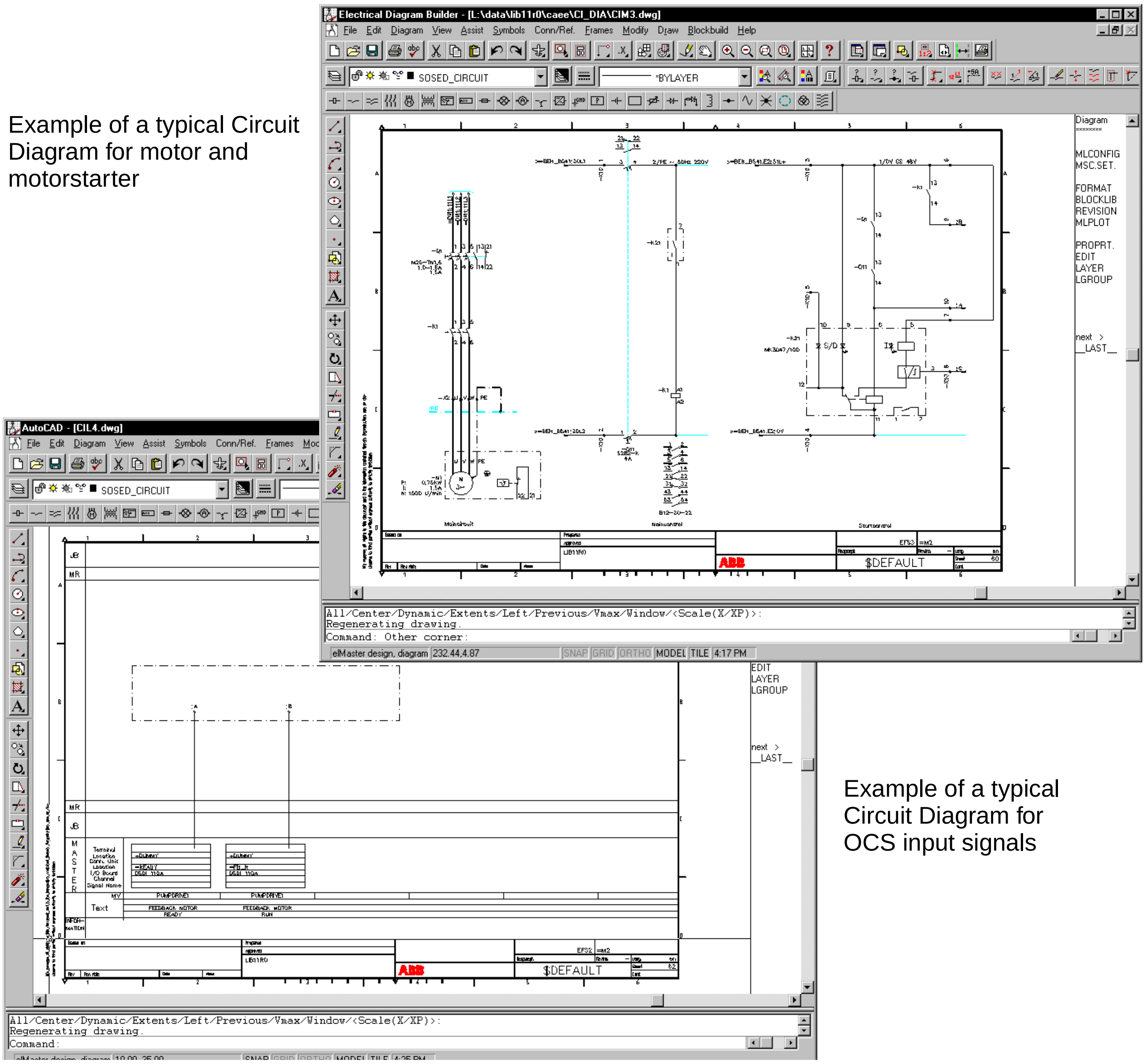
Electrical Cable Aspect

This aspect is inserted in objects that represent electrical cables to which cable data shall be assigned.

Electrical Circuit Diagram Aspect

Electrical Circuit Diagrams in a library contain typicals that are completed and interconnected at the application in which they will be used. In the standard pump drive system of our example we have several typical Circuit Diagrams, one for the main motor circuit and the motorstarter logic, one for the OCS Input Signals Circuit Diagram and one for the OCS Output Signals Circuit Diagram. Figure S6-14 shows the typical Circuit Diagrams for the main motor circuit and the OCS input signals circuits. The container for the typicals in our example are the Electrical Circuit Diagram aspects in Advant objects of type Binder.

Example of a typical Circuit Diagram for motor and motorstarter


Electrical Diagram Builder - [L:\data\lib11r0\caee\CI_DIA\CIM3.dwg]
File Edit Diagram View Assist Symbols Conn/Ref. Frames Modify Draw Blockbuild Help
SOSED_CIRCUIT
BYLAYER
Diagram
MLCONFIG
MSC.SET.
FORMAT
BLOCKLIB
REVISION
MLPLOT
PROPRT.
EDIT
LAYER
LGROUP
next >
__LAST__
$DEFAULT
elMaster design, diagram 232.44,4.87
SNAP GRID ORTHO MODEL TILE 4:17 PM
AutoCAD - [CIL4.dwg]
File Edit Diagram View Assist Symbols Conn/Ref. Frames Moc
SOSED_CIRCUIT
EDIT
LAYER
LGROUP
next >
__LAST__
PUMPDRIVE
FEEDBACK MOTOR READY
FEEDBACK MOTOR RUN
$DEFAULT

```
All/Center/Dynamic/Extents/Left/Previous/Vmax/Window/<Scale(X/XP)>:
Regenerating drawing.
Command: Other corner:
```

```
All/Center/Dynamic/Extents/Left/Previous/Vmax/Window/<Scale(X/XP)>:
Regenerating drawing.
Command:
```

Example of a typical Circuit Diagram for OCS input signals

Figure S6-14. Typicals of Electrical Circuit Diagrams

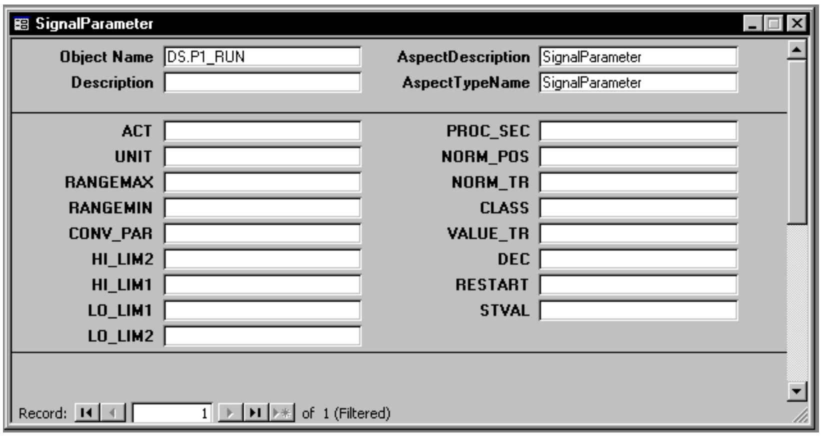
S6.3.2.2 OCS Related Aspects

Database Element Aspect

The DBElement aspect is assigned to the I/O signals of the standard drive system and references to a type circuit for the corresponding signal DB Element.

Parameter Aspect

The signal DB Elements are parametrized by values kept in Parameter aspects. The values can be input later in the application where the standard drive system is used. Figure S6-15 shows the predefined parameter aspect used for signals.



SignalParameter

Object Name: DS.P1_RUN
Description:
AspectDescription: SignalParameter
AspectTypeName: SignalParameter

Record: 1 of 1 (Filtered)

Figure S6-15. Signal Parameters

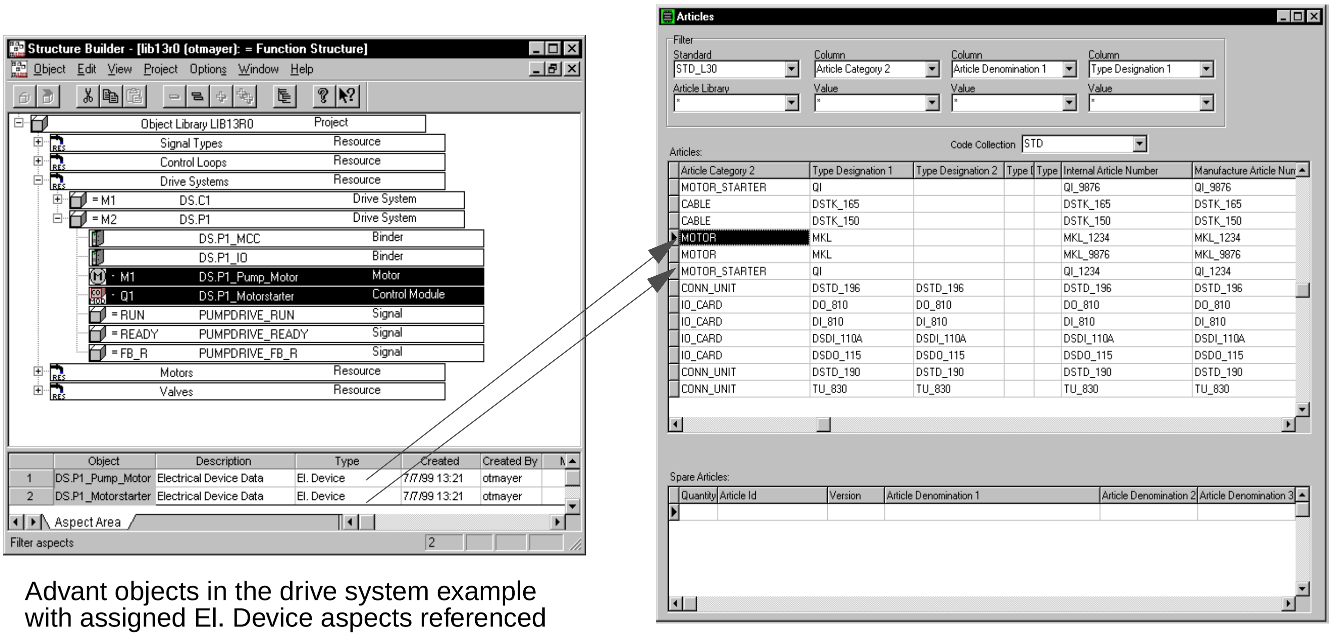
Circuit Aspect

The top object of the subtree representing the standard drive system would carry the circuit aspect with the control logic for the defined standard. This circuit aspect is not inserted in the presented example to keep the focus on the electrical engineering.

S6.3.3 Assigning Articles- (Product-) Data to Advant Object Aspects

Main articles are represented in the Advant object model by an Electrical Device aspect assigned to an Advant object. Electrical device aspects can be linked to articles defined in an article library. Precondition is that this article library has been assigned to the project. When an Advant object containing an electrical device aspect with a link to an article library is copied to another project, the article library will also be copied to this project.

Figure S6-16 shows the presentation of the main articles in our drive system example and the corresponding article data in the article library.


Structure Builder - [lib13r0 (otmayer): = Function Structure]
Object
Edit
View
Project
Options
Window
Help

Object Library LIB13R0 Project
Signal Types Resource
Control Loops Resource
Drive Systems Resource
= M1 DS.C1 Drive System
= M2 DS.P1 Drive System
DS.P1_MCC Binder
DS.P1_IO Binder
- M1 DS.P1_Pump_Motor Motor
- Q1 DS.P1_Motorstarter Control Module
= RUN PUMPDRIVE_RUN Signal
= READY PUMPDRIVE_READY Signal
= FB_R PUMPDRIVE_FB_R Signal
Motors Resource
Valves Resource

	Object	Description	Type	Created	Created By	M
1	DS.P1_Pump_Motor	Electrical Device Data	El. Device	7/7/99 13:21	otmayer	
2	DS.P1_Motorstarter	Electrical Device Data	El. Device	7/7/99 13:21	otmayer	

Advant objects in the drive system example with assigned El. Device aspects referenced to articles in the articles library

Standard: STD_L30
Article Library: *
Column: Article Category 2
Value: *
Column: Article Denomination 1
Value: *
Column: Type Designation 1
Value: *
Code Collection: STD

Articles:

Article Category 2	Type Designation 1	Type Designation 2	Type D	Type	Internal Article Number	Manufacture Article Num
MOTOR_STARTER	QI				QI_9876	QI_9876
CABLE	DSTK_165				DSTK_165	DSTK_165
CABLE	DSTK_150				DSTK_150	DSTK_150
MOTOR	MKL				MKL_1234	MKL_1234
MOTOR	MKL				MKL_9876	MKL_9876
MOTOR_STARTER	QI				QI_1234	QI_1234
CONN_UNIT	DSTD_196	DSTD_196			DSTD_196	DSTD_196
IO_CARD	DO_810	DO_810			DO_810	DO_810
IO_CARD	DI_810	DI_810			DI_810	DI_810
IO_CARD	DSDI_110A	DSDI_110A			DSDI_110A	DSDI_110A
IO_CARD	DSDO_115	DSDO_115			DSDO_115	DSDO_115
CONN_UNIT	DSTD_190	DSTD_190			DSTD_190	DSTD_190
CONN_UNIT	TU_830	TU_830			TU_830	TU_830

Spare Articles:

Quantity	Article Id	Version	Article Denomination 1	Article Denomination 2	Article Denomination 3

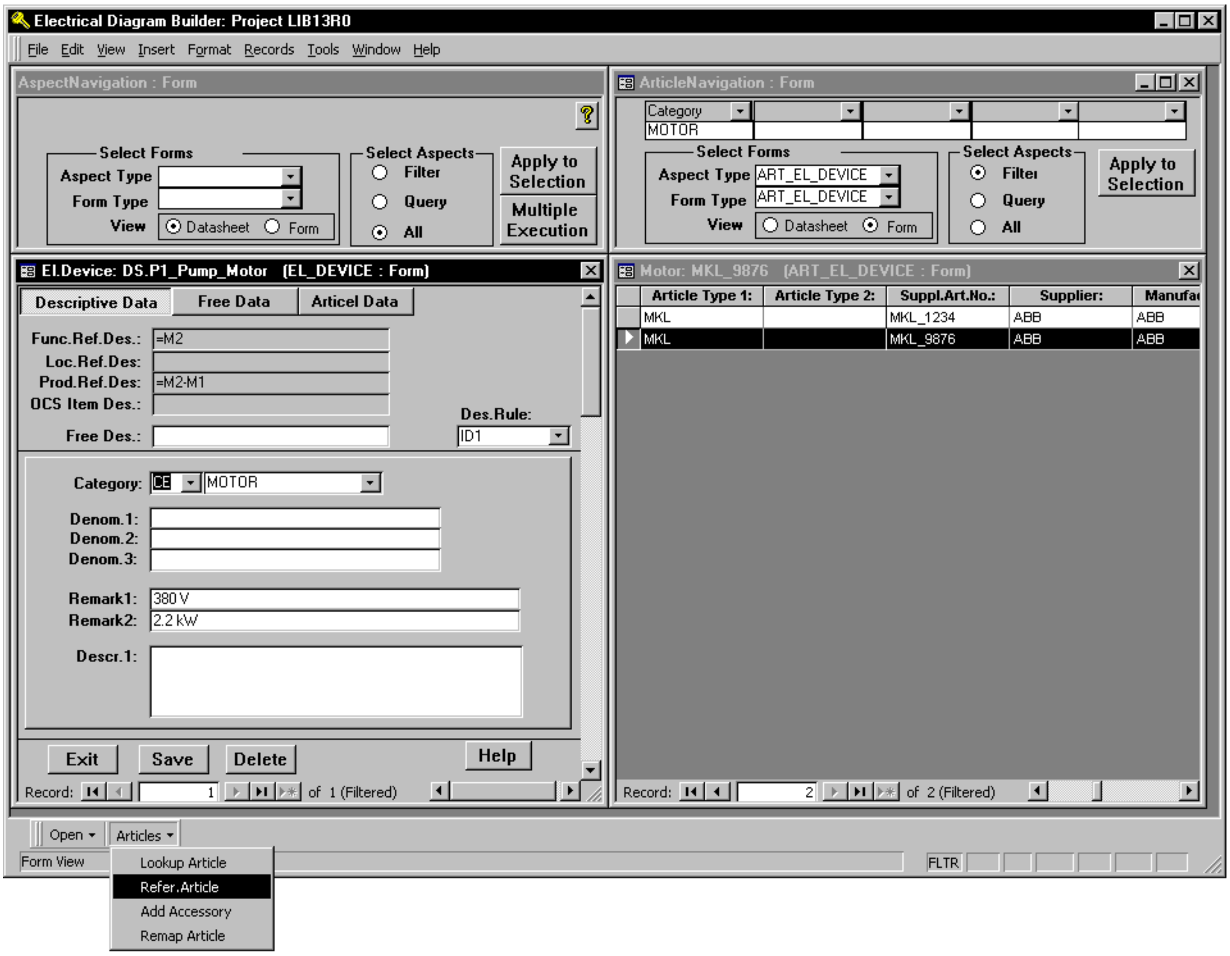
Contents of the Articles Library

Figure S6-16. Electrical Articles Assigned to El. Device Aspects

Creation of References Between Object Aspects and Articles

Open in the El. Diagram Builder the Attributes view of El. Device aspect and the relevant data sheet view of the articles library as shown in Figure S6-17 to perform a reference from an el. device aspect to an article.

Assign a Main Article to an Advant Object Aspect	
Preconditions:	Article library assigned to object library and Advant object created to which article shall be assigned
User activities:	1. Start up the Structure Builder 2. Select the object to which a main article shall be assigned 3. Open the object and open the **Attributes** view of the **El. Device** aspect 4. Choose **Articles | Lookup Article** in the toolbar of the **Electrical Diagram Builder** window: 5. Navigate to articles by the **Article Navigation** Form and the **ART_EL_DEVICE** Form 6. Select the article you want to reference in the **ART_EL_DEVICE** Form 7. Select the el. device the reference shall be assigned to for example by clicking on the left bar in the open **EL_DEVICE** Form 8. Choose **Articles | Refer. Article** in the toolbar of the **Electrical Diagram Builder** window 9. Inspect the article data on page 3 of the **EL_DEVICE** Form and complete the information input to the **EL_DEVICE** Form 10. Choose **Exit** from the **File** menu of the Electrical Diagram Builder**.**
Result:	Article assigned to El. Device aspect of Advant object



AspectNavigation : Form

Aspect Type:
Form Type:
View: ● Datasheet ○ Form

Select Aspects: ○ Filter ○ Query ● All

ArticleNavigation : Form

Category: MOTOR

Aspect Type: ART_EL_DEVICE
Form Type: ART_EL_DEVICE
View: ○ Datasheet ● Form

Select Aspects: ● Filter ○ Query ○ All

El.Device: DS.P1_Pump_Motor (EL_DEVICE : Form)

Func.Ref.Des.: =M2
Loc.Ref.Des:
Prod.Ref.Des: =M2-M1
OCS Item Des.:
Free Des.:
Des.Rule: ID1

Category: CE MOTOR

Denom.1:
Denom.2:
Denom.3:

Remark1: 380 V
Remark2: 2.2 kW

Descr.1:

Record: 1 of 1 (Filtered)

Motor: MKL_9876 (ART_EL_DEVICE : Form)

Article Type 1:	Article Type 2:	Suppl.Art.No.:	Supplier:	Manufa
MKL		MKL_1234	ABB	ABB
MKL		MKL_9876	ABB	ABB

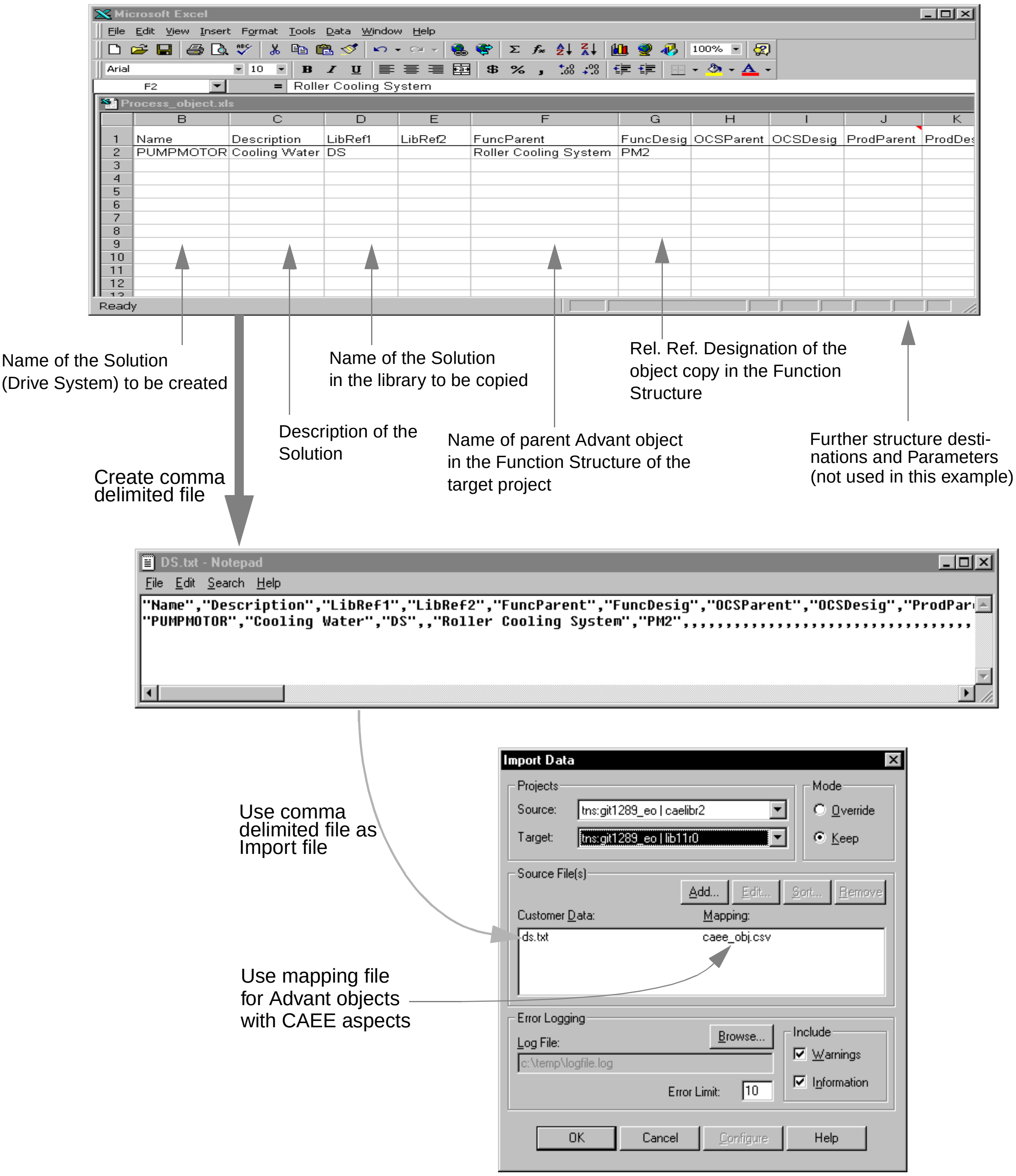
Record: 2 of 2 (Filtered)

Figure S6-17. Referencing Article Data

S6.4 Importing Standard Solutions into a Library

Solutions can be inserted into a project specific object library by developing it from scratch but solutions can also be inserted by importing it from other libraries or from application projects. This is performed by using the import functionality of the Structure Builder. This is illustrated in Figure S6-18 where we see the procedure for the import of a solution which is in this case the drive system of our example.

Import a Standard Solution for a Drive System	
Preconditions:	Solution available in an existing library
User activities:	1. Create the import file in MS Access or MS Excel 2. Convert the import file to a comma delimited file 3. Use the Structure Builder and choose Import from the Project menu 4. Insert source project, target project, import data file, mapping file and logfile into the Import Data window 5. Click on OK
Result:	Standard Solution used from a source project (solution library or application project) and inserted into the target project (project library)



F2 = Roller Cooling System

	B	C	D	E	F	G	H	I	J	K
1	Name	Description	LibRef1	LibRef2	FuncParent	FuncDesig	OCSParent	OCSDesig	ProdParent	ProdDes
2	PUMPMOTOR	Cooling Water	DS		Roller Cooling System	PM2				

Name of the Solution (Drive System) to be created

Description of the Solution

Name of the Solution in the library to be copied

Name of parent Advant object in the Function Structure of the target project

Rel. Ref. Designation of the object copy in the Function Structure

Further structure destinations and Parameters (not used in this example)

Create comma delimited file

```
"Name","Description","LibRef1","LibRef2","FuncParent","FuncDesig","OCSParent","OCSDesig","ProdPar
"PUMPMOTOR","Cooling Water","DS",,"Roller Cooling System","PM2",,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
```

Use comma delimited file as Import file

Use mapping file for Advant objects with CAEE aspects

Projects
Source: tns:git1289_eo | caelibr2
Target: tns:git1289_eo | lib11r0

Mode
○ Override
● Keep

Source File(s)
Customer Data: ds.txt
Mapping: caee_obj.csv

Error Logging
Log File: c:\temp\logfile.log
Error Limit: 10

Include
☑ Warnings
☑ Information

Figure S6-18. Importing a Solution

S6.5 Using Standard Solutions

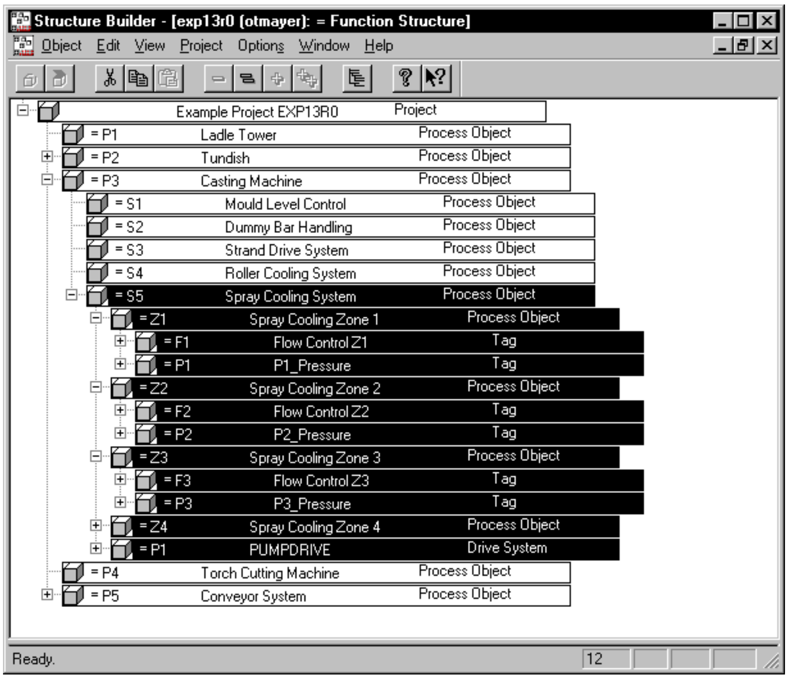
S6.5.1 Copying and Importing Standard Solutions to Application Projects

The spray cooling system in our example consists of four cooling zones and a drive system for the cooling water pump. Each zone contains a flow control loop and a pressure monitoring system.

We have built up our Function Oriented Structure down to the individual cooling zones. Now we use our standards from the library project. Copy the standard flow control loop under each of the four spray cooling zones. Copy the drive system and place it below the Advant object spray cooling system.

<table>
<tr><td colspan="2">Copying a Substructure from the Library Project
into the Function Structure of a Customer Project
(for example a standard control loop or a standard drive system)</td></tr>
<tr><td><i>Preconditions:</i></td><td>Standard solutions in the library project available.
Basic Function Structure of application project created</td></tr>
<tr><td><i>User activities:</i></td><td>1. Use the Structure Builder
2. Select the object on top of the standard solution that shall be copied
3. Choose Copy in the Edit menu
4. Click on the object in the Function Structure to which the subtree of the control loop shall be copied -either as child or above/below the selected object on same level.
5. Choose Paste Special in the Edit menu
6. Select Copy and With Subtree in the Paste Special window
7. Select one of the placing items
– as child
– above on same level
– below on same level
8. Click on OK
9. Insert new name(s) for copied Circuit(s)
10. Click on OK</td></tr>
<tr><td><i>Result:</i></td><td>Standard solution copied and placed in the Function Structure</td></tr>
</table>

Figure S6-19 shows the complete subtree of the spray cooling system in the application project.


Structure Builder - [exp13r0 (otmayer): = Function Structure]
Object
Edit
View
Project
Options
Window
Help
Ready.
12

```
Example Project EXP13R0              Project
  = P1   Ladle Tower                 Process Object
  = P2   Tundish                     Process Object
  = P3   Casting Machine             Process Object
    = S1   Mould Level Control       Process Object
    = S2   Dummy Bar Handling        Process Object
    = S3   Strand Drive System       Process Object
    = S4   Roller Cooling System     Process Object
    = S5   Spray Cooling System      Process Object
      = Z1   Spray Cooling Zone 1    Process Object
        = F1   Flow Control Z1       Tag
        = P1   P1_Pressure           Tag
      = Z2   Spray Cooling Zone 2    Process Object
        = F2   Flow Control Z2       Tag
        = P2   P2_Pressure           Tag
      = Z3   Spray Cooling Zone 3    Process Object
        = F3   Flow Control Z3       Tag
        = P3   P3_Pressure           Tag
      = Z4   Spray Cooling Zone 4    Process Object
      = P1   PUMPDRIVE               Drive System
  = P4   Torch Cutting Machine       Process Object
  = P5   Conveyor System             Process Object
```

Figure S6-19. Example for the Solution of a Cooling System

By copying Advant objects also all aspects and aspect data are copied. Missing aspect data must be inserted and/or predefined parameters must be set to instance specific values when standard solutions are instantiated.

A more efficient way of using standard solutions is by applying the import functionality of the AdvaBuild Engineering Software instead of copying solutions manually. The handling of this import functionality is described in Step 7, Step 8, Step 9, and Step 10.

S6.5.2 Placing the Objects of a Solution in the Application Project

The drive system that we created in the object library is used in an application. The function to which the drive system shall belong as subfunction should have been defined and placed in the Function Structure of the application project before the standard solution is used.

The complete drive system as defined in the library can then be hooked to this function. All objects will be in the Function Structure of the application project but some objects need additional placements into other structures as for example into the Mounting Location Structure and into the OCS-Structure. Table 6-1 gives an overview where the objects of the drive system of our example will be placed in the further planning steps.

Object	**Function Structure**	**OCS-Structure**	**Mounting Location Structure**	**Location Structure**
DS.P1 (Drive System)	+			
DS.P1_MCC (Binder)	+		+	
DS.P1_IO (Binder)	+		+	
DS.P1_Pump_Motor	+			+
DS.P1_Motorstarter	+		+	
DS.P1_RUN (I/O Signal)	+	+		
DS.P1_READY (I/O Signal)	+	+		
DS.P1_FB_RUN (I/O Signal)	+	+		

Table S6-1. Typical Placements of Solution Objects in an Application

Placements in the OCS-Structure

The input/output signals will be allocated to board channels in the OCS-Structure. The board channel objects are assigned to board objects. Board objects are placed below node objects representing the functionality of a specific controller. Placing the signal objects can be performed via drag and drop below a channel object or by using the import functionality of the Structure Builder.

Precondition for placing the objects in the OCS-Structure is the existence of board types that match the signal types.

The object name of a signal object will become the signal name in the OCS-system. The relative reference designation of a signal will not be evaluated by the OCS-system, only the channel designation concatenated with the board designation will be evaluated.

Placements in the Mounting Location Structure

Mounting locations are racks, cabinets, enclosures and so on that get a unique reference designation within a plant. The motorstarter of our example can be placed in such a mounting location, in this case in a cabinet or rack for motorstarters. The reference designation of the mounting location where the motorstarter is placed shall be inserted on the title block of the Circuit Diagrams. This is done automatically by the functionality of the Electrical Diagram Builder if the object carrying the diagrams is placed as child to this mounting location. The Circuit Diagrams are therefore assigned to Binder objects which are placed below the mounting location (cabinet) that carries the motorstarter.

The Circuit Diagrams for the input/output signal paths to the OCS-system should get the reference designation of the cabinet where the boards the signal are assigned to are located. The

Binder object with the Input/Output Signal Circuit Diagrams is therefore assigned to the cabinet in the Mounting Location Structure that contains the corresponding boards.

Placements in the (Topographic) Location Structure

Objects representing components that need no special enclosures (mounting locations) are placed directly into the Topographic Location Structure. This will be done for the motor object in our example.

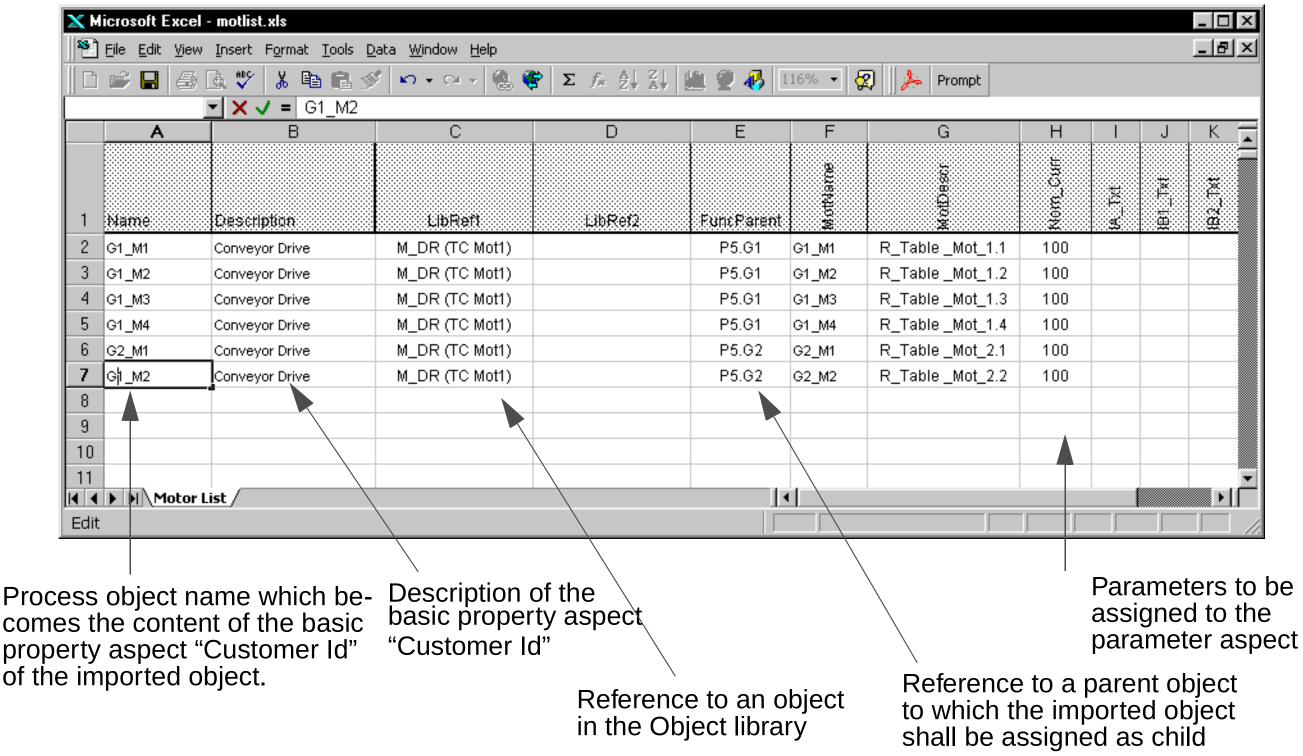
Step 7 Build Process Object Data

S7.1 Creating the Process Object List

You can define Process objects as for example motors and consumers with its instance specific parameters in a process object import list. The items in such a list represent objects that shall be imported into the Advant object model of a project. The listed objects refer to objects in an Object library project representing a standard solution. Each row in the process object list points to an object in the library project which will be copied with its subobjects into the project model during the import process. Such a library object may contain beside other aspects a Circuit aspect and a Parameter aspect. The parameter aspect of an imported object is filled with the given parameters of the corresponding item in the process object list.

The process object list can be handled by Excel. For its use for the final import it has to be converted into a comma delimited file.

Figure S7-1 shows an example of a process object list. Figure S7-2 the corresponding comma delimited file used for the import procedure.


Microsoft Excel - motlist.xls
File Edit View Insert Format Tools Data Window Help
116%
Prompt
G1_M2
Motor List
Edit

	A	B	C	D	E	F	G	H	I	J	K
1	Name	Description	LibRef1	LibRef2	FuncParent	MotName	MotDescr	Nom_Curr	IA_Txt	IB1_Txt	IB2_Txt
2	G1_M1	Conveyor Drive	M_DR (TC Mot1)		P5.G1	G1_M1	R_Table _Mot_1.1	100			
3	G1_M2	Conveyor Drive	M_DR (TC Mot1)		P5.G1	G1_M2	R_Table _Mot_1.2	100			
4	G1_M3	Conveyor Drive	M_DR (TC Mot1)		P5.G1	G1_M3	R_Table _Mot_1.3	100			
5	G1_M4	Conveyor Drive	M_DR (TC Mot1)		P5.G1	G1_M4	R_Table _Mot_1.4	100			
6	G2_M1	Conveyor Drive	M_DR (TC Mot1)		P5.G2	G2_M1	R_Table _Mot_2.1	100			
7	G1_M2	Conveyor Drive	M_DR (TC Mot1)		P5.G2	G2_M2	R_Table _Mot_2.2	100			
8											
9											
10											
11											

Process object name which becomes the content of the basic property aspect "Customer Id" of the imported object.

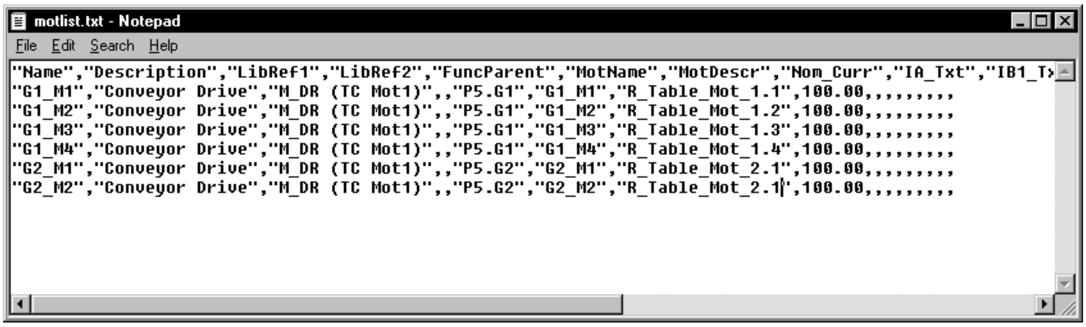
Description of the basic property aspect "Customer Id"

Reference to an object in the Object library

Reference to a parent object to which the imported object shall be assigned as child

Parameters to be assigned to the parameter aspect

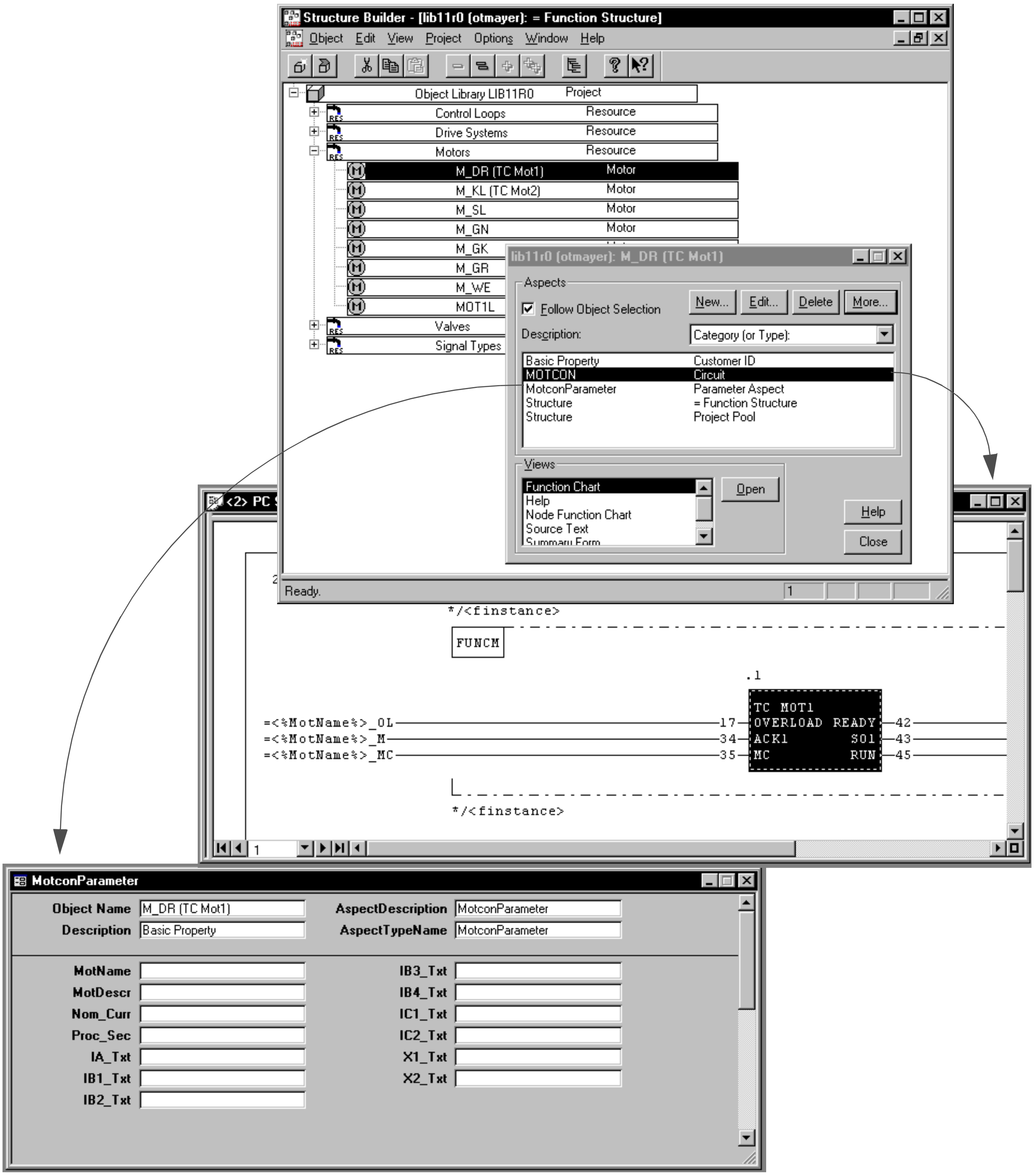
Figure S7-1. Process Object List



```
"Name","Description","LibRef1","LibRef2","FuncParent","MotName","MotDescr","Nom_Curr","IA_Txt","IB1_T
"G1_M1","Conveyor Drive","M_DR (TC Mot1)",,"P5.G1","G1_M1","R_Table_Mot_1.1",100.00,,,,,,,,,
"G1_M2","Conveyor Drive","M_DR (TC Mot1)",,"P5.G1","G1_M2","R_Table_Mot_1.2",100.00,,,,,,,,,
"G1_M3","Conveyor Drive","M_DR (TC Mot1)",,"P5.G1","G1_M3","R_Table_Mot_1.3",100.00,,,,,,,,,
"G1_M4","Conveyor Drive","M_DR (TC Mot1)",,"P5.G1","G1_M4","R_Table_Mot_1.4",100.00,,,,,,,,,
"G2_M1","Conveyor Drive","M_DR (TC Mot1)",,"P5.G2","G2_M1","R_Table_Mot_2.1",100.00,,,,,,,,,
"G2_M2","Conveyor Drive","M_DR (TC Mot1)",,"P5.G2","G2_M2","R_Table_Mot_2.1",100.00,,,,,,,,,
```

Figure S7-2. Comma Delimited Format of a Process Object List

Figure S7-3 shows the process object in the Object library which was selected as reusable solution and to which is referenced in the import list.


Structure Builder - [lib11r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
Ready.
1
lib11r0 (otmayer): M_DR (TC Mot1)
Aspects
New...
Edit...
Delete
More...
Views
Open
Help
Close
<2> PC
MotconParameter

Object Library LIB11R0	Project
Control Loops	Resource
Drive Systems	Resource
Motors	Resource
M_DR (TC Mot1)	Motor
M_KL (TC Mot2)	Motor
M_SL	Motor
M_GN	Motor
M_GK	
M_GR	
M_WE	
MOT1L	
Valves	
Signal Types	

☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
MOTCON	Circuit
MotconParameter	Parameter Aspect
Structure	= Function Structure
Structure	Project Pool

Views: Function Chart, Help, Node Function Chart, Source Text, Summary Form

```
*/<finstance>
FUNCM
                                              .1
                                            TC MOT1
=<%MotName%>_OL-------------------------17-OVERLOAD READY-42
=<%MotName%>_M--------------------------34-ACK1      S01-43
=<%MotName%>_MC-------------------------35-MC        RUN-45
*/<finstance>
```

Object Name	M_DR (TC Mot1)	**AspectDescription**	MotconParameter
Description	Basic Property	**AspectTypeName**	MotconParameter
MotName		**IB3_Txt**	
MotDescr		**IB4_Txt**	
Nom_Curr		**IC1_Txt**	
Proc_Sec		**IC2_Txt**	
IA_Txt		**X1_Txt**	
IB1_Txt		**X2_Txt**	
IB2_Txt			

Figure S7-3. Process Object in the Object Library

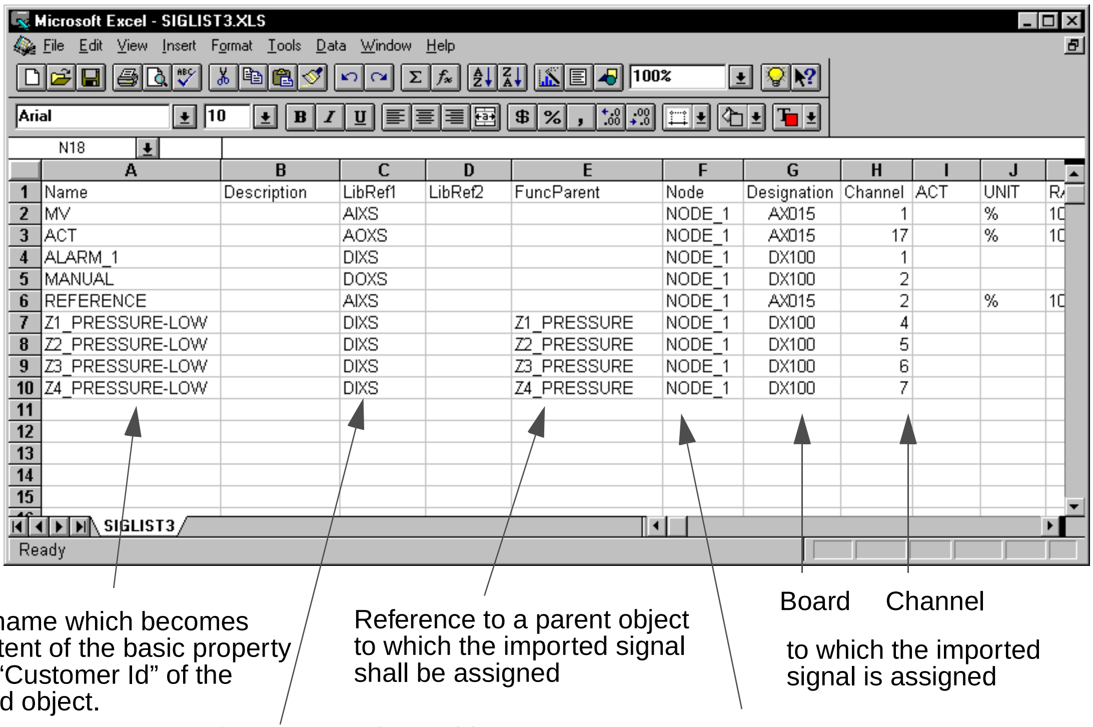
Step 8 Build Signal Data

S8.1 Creating the Signal List

You can define signals for the import into the object model of a project in a signal list. The signals are listed with a reference to objects in an Object library representing standard signals. Each row in the signal list leads to a reference object which is copied into the project model during the import process. Each reference object contains beside other aspects a DB-element aspect and a Signal Parameter aspect. The parameter aspect of an imported signal is filled with the given parameters of the corresponding signal list item.

You can handle signal lists by Excel. For its use for the final import procedure it is converted into a comma delimited file.

Figure S8-1 show an example of a signal list that we use in our example. Figure S8-2 the corresponding comma delimited file used for the import procedure.



	A	B	C	D	E	F	G	H	I	J	R
1	Name	Description	LibRef1	LibRef2	FuncParent	Node	Designation	Channel	ACT	UNIT	R
2	MV		AIXS			NODE_1	AX015	1		%	1
3	ACT		AOXS			NODE_1	AX015	17		%	1
4	ALARM_1		DIXS			NODE_1	DX100	1			
5	MANUAL		DOXS			NODE_1	DX100	2			
6	REFERENCE		AIXS			NODE_1	AX015	2		%	1
7	Z1_PRESSURE-LOW		DIXS		Z1_PRESSURE	NODE_1	DX100	4			
8	Z2_PRESSURE-LOW		DIXS		Z2_PRESSURE	NODE_1	DX100	5			
9	Z3_PRESSURE-LOW		DIXS		Z3_PRESSURE	NODE_1	DX100	6			
10	Z4_PRESSURE-LOW		DIXS		Z4_PRESSURE	NODE_1	DX100	7			
11											
12											
13											
14											
15											

Signal name which becomes the content of the basic property aspect “Customer Id” of the imported object.

Reference to a signal object in the Object library

Reference to a parent object to which the imported signal shall be assigned

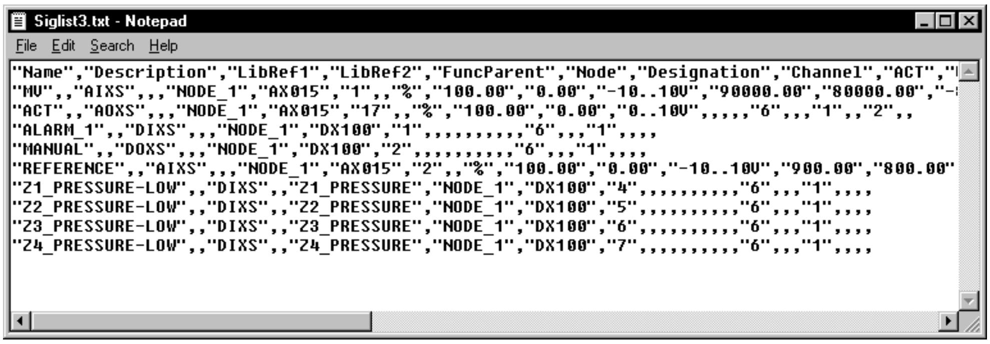
Node object in the OCS-Structure to which the imported signal object is assigned

Board   Channel

to which the imported signal is assigned

All the other information within one row are assigned as parameters to the parameter aspect of the imported signal object

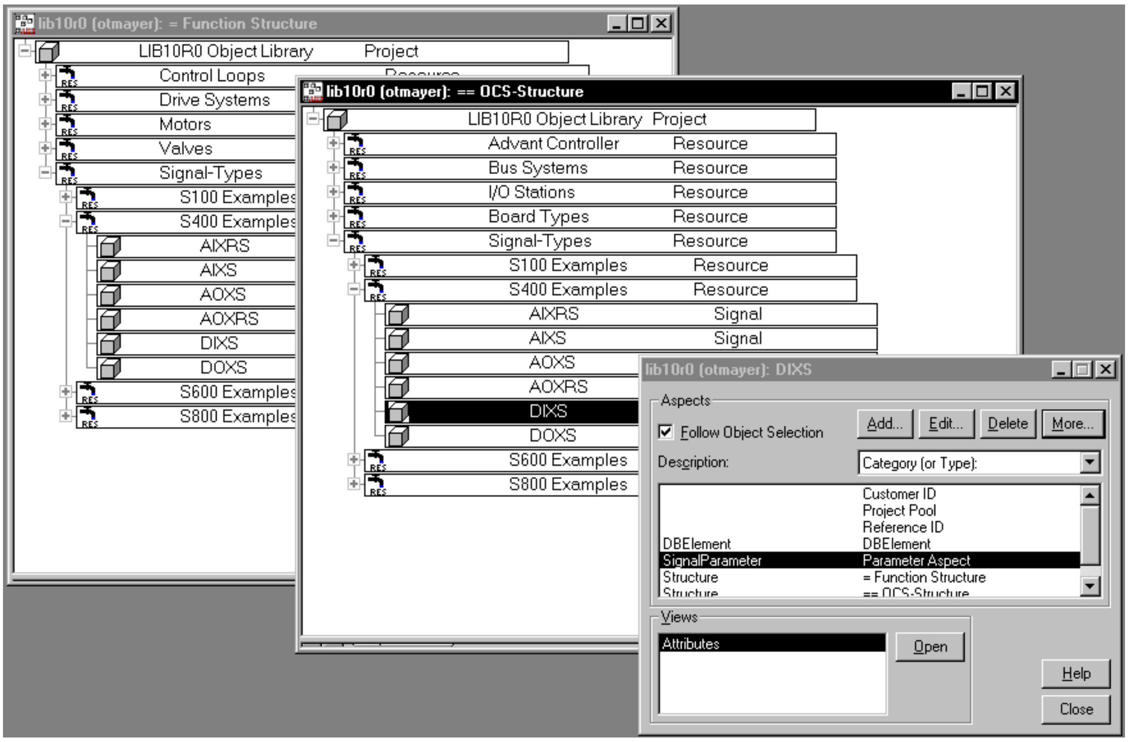
Figure S8-1. Signal List



```
"Name","Description","LibRef1","LibRef2","FuncParent","Node","Designation","Channel","ACT","
"MV",,"AIXS",,,"NODE_1","AX015","1",,"%","100.00","0.00","-10..10V","90000.00","80000.00","-
"ACT",,"AOXS",,,"NODE_1","AX015","17",,"%","100.00","0.00","0..10V",,,,,"6",,,"1",,"2",,
"ALARM_1",,"DIXS",,,"NODE_1","DX100","1",,,,,,,,,,"6",,,"1",,,,
"MANUAL",,"DOXS",,,"NODE_1","DX100","2",,,,,,,,,,"6",,,"1",,,,
"REFERENCE",,"AIXS",,,"NODE_1","AX015","2",,"%","100.00","0.00","-10..10V","900.00","800.00"
"Z1_PRESSURE-LOW",,"DIXS",,"Z1_PRESSURE","NODE_1","DX100","4",,,,,,,,,,"6",,,"1",,,,
"Z2_PRESSURE-LOW",,"DIXS",,"Z2_PRESSURE","NODE_1","DX100","5",,,,,,,,,,"6",,,"1",,,,
"Z3_PRESSURE-LOW",,"DIXS",,"Z3_PRESSURE","NODE_1","DX100","6",,,,,,,,,,"6",,,"1",,,,
"Z4_PRESSURE-LOW",,"DIXS",,"Z4_PRESSURE","NODE_1","DX100","7",,,,,,,,,,"6",,,"1",,,,
```

Figure S8-2. Comma Delimited Format of a Signal List

Figure S8-3 shows the predefined signal object in the library to which is referenced in the import list. The signals to be imported shall be placed in the target project in the Function Structure and in the Control System Structure and must therefore also be placed in the library in this two structures.



☑ Follow Object Selection

Description: Category (or Type):

	Customer ID
	Project Pool
	Reference ID
DBElement	DBElement
SignalParameter	Parameter Aspect
Structure	= Function Structure
Structure	== OCS-Structure

Views: Attributes

Figure S8-3. Signal Object in the Object Library Placed in two Structures

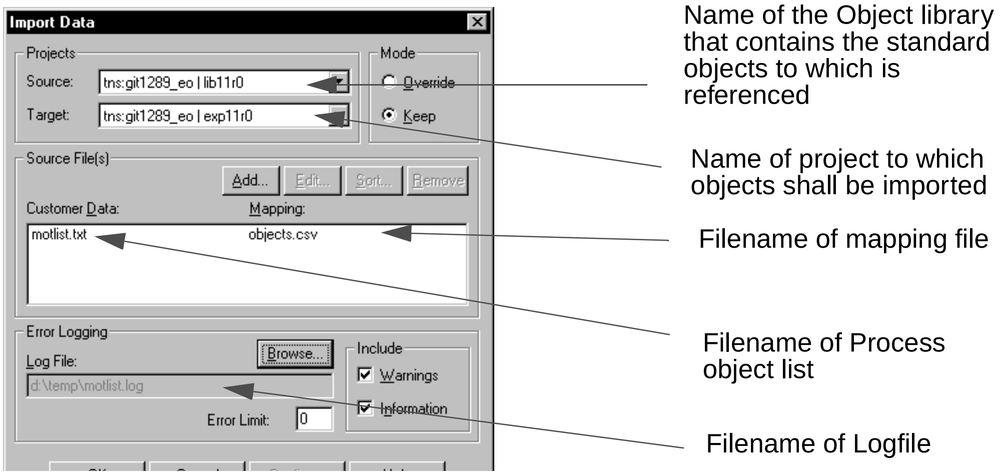
Step 9 Import of Process Object Data

S9.1 Import Procedure

Preconditions for the start of the import procedure for process objects are:

- Object list with the objects to be imported is available as comma delimited file
- Object library project is available with all the object types defined to which is referenced.
- Object types in the Object library project must have the structure aspects of the destination structures according the import lists.

Import of Process Object Data	
User activities:	1. Use the Structure Builder 2. Choose **Import** from the **Project** menu 3. Select the source project (Object library) and the target project in the **Import Data** window 4. Insert the Process object list file, mapping file and logfile 5. Click on **OK**
Result:	Process object and its assigned data imported to project



Source: tns:git1289_eo | lib11r0
Target: tns:git1289_eo | exp11r0
○ Override
● Keep
Customer Data: motlist.txt
Mapping: objects.csv
Log File: d:\temp\motlist.log
Error Limit: 0
☑ Warnings
☑ Information

Name of the Object library that contains the standard objects to which is referenced

Name of project to which objects shall be imported

Filename of mapping file

Filename of Process object list

Filename of Logfile

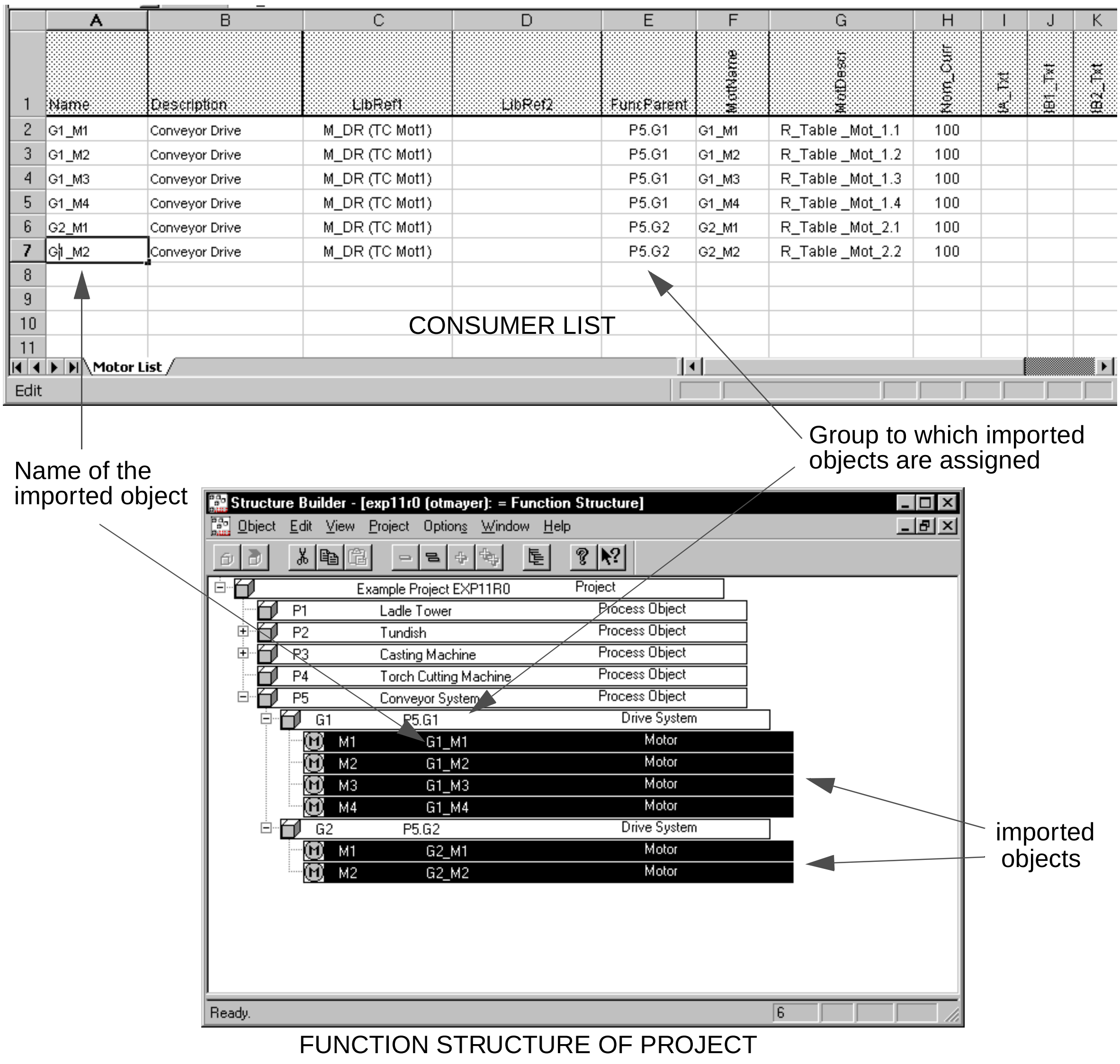
Figure S9-1. Import Dialog

Answer the dialog boxes during the import procedure and check the logfile after the import.

S9.2 Import Results

S9.2.1 Creation of Process Objects in the Target Project

The items of the process object list are placed by the import procedure into the Function Structure of the project. If a parent reference to an object in the Function Structure of the project is given, the instance will be placed as child below this object. Object without a destination reference will be put as root objects into the Function Structure and must be placed manually. In our example we placed several motor objects via import function into two groups of drive systems of a conveyor system.



	A	B	C	D	E	F	G	H	I	J	K
1	Name	Description	LibRef1	LibRef2	FuncParent	MotName	MotDescr	Nom_Curr	IA_Txt	IB1_Txt	IB2_Txt
2	G1_M1	Conveyor Drive	M_DR (TC Mot1)		P5.G1	G1_M1	R_Table _Mot_1.1	100			
3	G1_M2	Conveyor Drive	M_DR (TC Mot1)		P5.G1	G1_M2	R_Table _Mot_1.2	100			
4	G1_M3	Conveyor Drive	M_DR (TC Mot1)		P5.G1	G1_M3	R_Table _Mot_1.3	100			
5	G1_M4	Conveyor Drive	M_DR (TC Mot1)		P5.G1	G1_M4	R_Table _Mot_1.4	100			
6	G2_M1	Conveyor Drive	M_DR (TC Mot1)		P5.G2	G2_M1	R_Table _Mot_2.1	100			
7	G1_M2	Conveyor Drive	M_DR (TC Mot1)		P5.G2	G2_M2	R_Table _Mot_2.2	100			
8											
9											
10											
11											

Example Project EXP11R0 Project
- P1 Ladle Tower Process Object
- P2 Tundish Process Object
- P3 Casting Machine Process Object
- P4 Torch Cutting Machine Process Object
- P5 Conveyor System Process Object
  - G1 P5.G1 Drive System
    - M1 G1_M1 Motor
    - M2 G1_M2 Motor
    - M3 G1_M3 Motor
    - M4 G1_M4 Motor
  - G2 P5.G2 Drive System
    - M1 G2_M1 Motor
    - M2 G2_M2 Motor

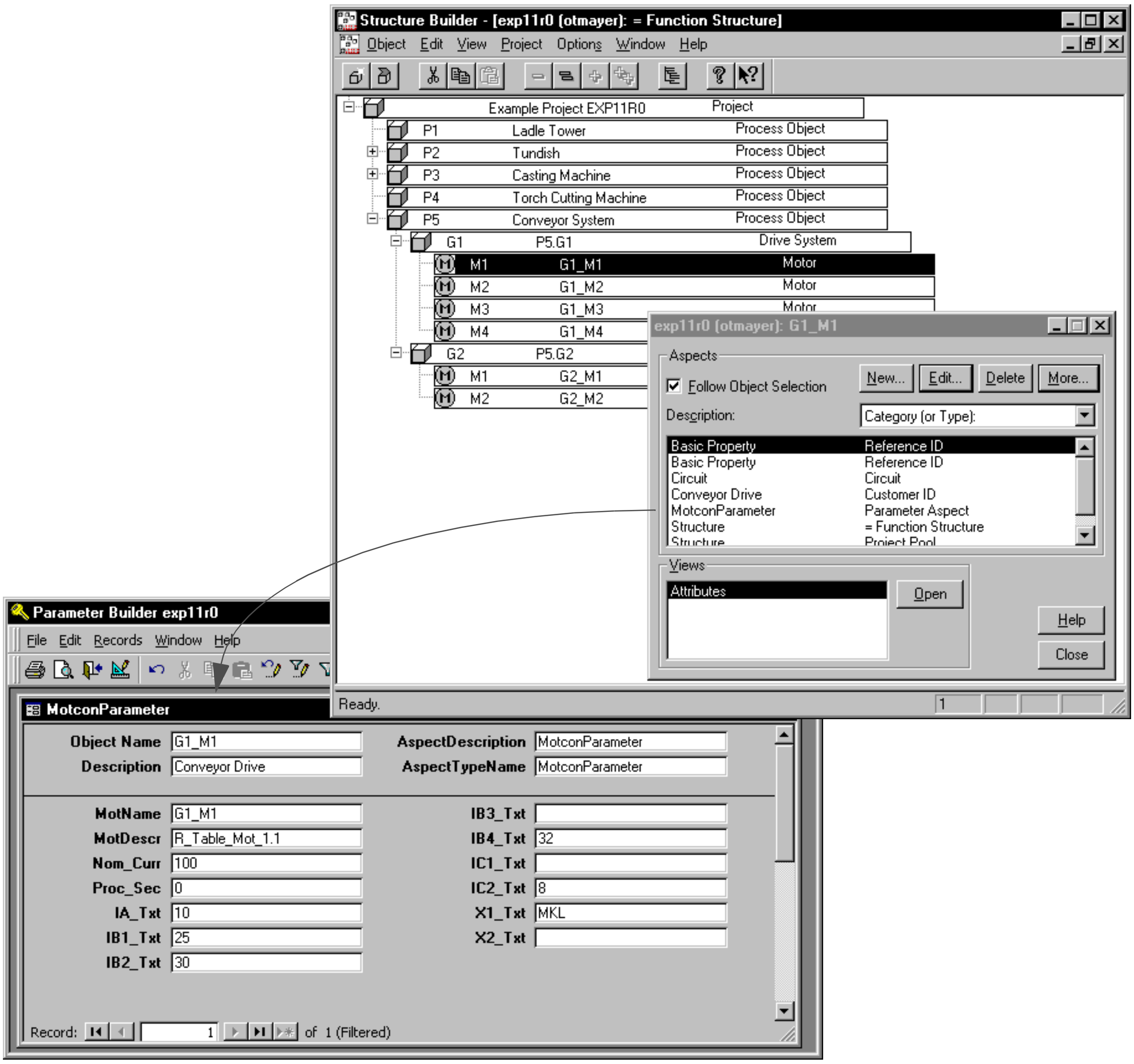
Motor List
Edit
CONSUMER LIST
Name of the imported object
Group to which imported objects are assigned
Structure Builder - [exp11r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
Ready.
6
imported objects
FUNCTION STRUCTURE OF PROJECT

Figure S9-2. Import of Process Objects

S9.2.2 Aspects and Parameters of Imported Process Objects

An instance of a process object gets all the aspects of the corresponding library object. The parameters in the process object list are put into the Parameter aspect of the object instance.

Figure S9-3 shows the aspects of an imported process object. Figure S9-3 shows the view on the Process Object Parameter aspect. The parameters are taken over from the process object list but can be edited afterwards by using this view.


Structure Builder - [exp11r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
Ready.
1
exp11r0 (otmayer): G1_M1
Aspects
New...
Edit...
Delete
More...
Views
Attributes
Open
Help
Close
Parameter Builder exp11r0
File Edit Records Window Help
MotconParameter
Record:
of 1 (Filtered)

	Example Project EXP11R0	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
P3	Casting Machine	Process Object
P4	Torch Cutting Machine	Process Object
P5	Conveyor System	Process Object
G1	P5.G1	Drive System
M1	G1_M1	Motor
M2	G1_M2	Motor
M3	G1_M3	Motor
M4	G1_M4	
G2	P5.G2	
M1	G2_M1	
M2	G2_M2	

☑ Follow Object Selection

Description:	Category (or Type):
Basic Property	Reference ID
Basic Property	Reference ID
Circuit	Circuit
Conveyor Drive	Customer ID
MotconParameter	Parameter Aspect
Structure	= Function Structure
Structure	Project Pool

Object Name: G1_M1
Description: Conveyor Drive
AspectDescription: MotconParameter
AspectTypeName: MotconParameter

MotName: G1_M1
MotDescr: R_Table_Mot_1.1
Nom_Curr: 100
Proc_Sec: 0
IA_Txt: 10
IB1_Txt: 25
IB2_Txt: 30
IB3_Txt:
IB4_Txt: 32
IC1_Txt:
IC2_Txt: 8
X1_Txt: MKL
X2_Txt:

Record: 1

Figure S9-3. Imported Process Object with View on its Parameter Aspect

The updating of the circuit aspect of an object instance by the contents of the Parameter aspect is a separate process. It can be done individually for each object or it can be done by one command for all relevant objects in the project. For more details see description of Step 16.

S9.2.3 Reimport of Process Object Data

You can perform a reimport of process object data from a process object list that has already been imported. This is usually done after modifications or further completion of the data in the process object list. You can perform reimport either in the keep mode or in the overwrite mode. In the keep mode an already existing aspect of a process object is kept with its former data. In the overwrite mode an already existing aspect of a process object is deleted and replaced by the data from the process object list used for reimport.

NOTE

The identity of an aspect is evaluated from the aspect type and the aspect description. If you change an aspect description within a process object and you perform a reimport with the old aspect description still in the process object list, this aspect will be handled like an additional aspect in the import result. Therefore aspect descriptions must not be changed if a reimport is performed.

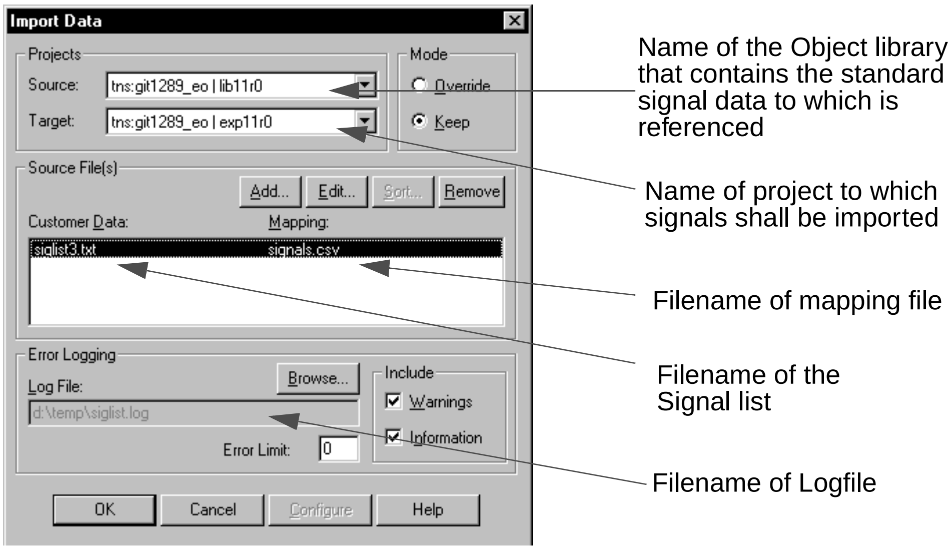
Step 10 Import of Signal Data

S10.1 Import Procedure

Preconditions for the start of the import procedure for signals are

- Signal list with the signals to be imported is available as comma delimited file
- Library project is available with all the signal types defined to which is referenced.
- Signal types in the Object library must have the structure aspects of the destination structures according the import lists.

Import of Signal Data	
User activities:	1. Use the Structure Builder 2. Choose **Import** from the **Project** menu 3. Select the source project (Object library) and the target project in the **Import Data** window 4. Insert the Signal list file, the mapping file and the logfile 5. Click on **OK**
Result:	Signals and its assigned data imported to project



Source: tns:git1289_eo | lib11r0
Target: tns:git1289_eo | exp11r0
○ Override
● Keep
Customer Data: siglist3.txt
Mapping: signals.csv
Log File: d:\temp\siglist.log
☑ Warnings
☑ Information
Error Limit: 0

Name of the Object library that contains the standard signal data to which is referenced

Name of project to which signals shall be imported

Filename of mapping file

Filename of the Signal list

Filename of Logfile

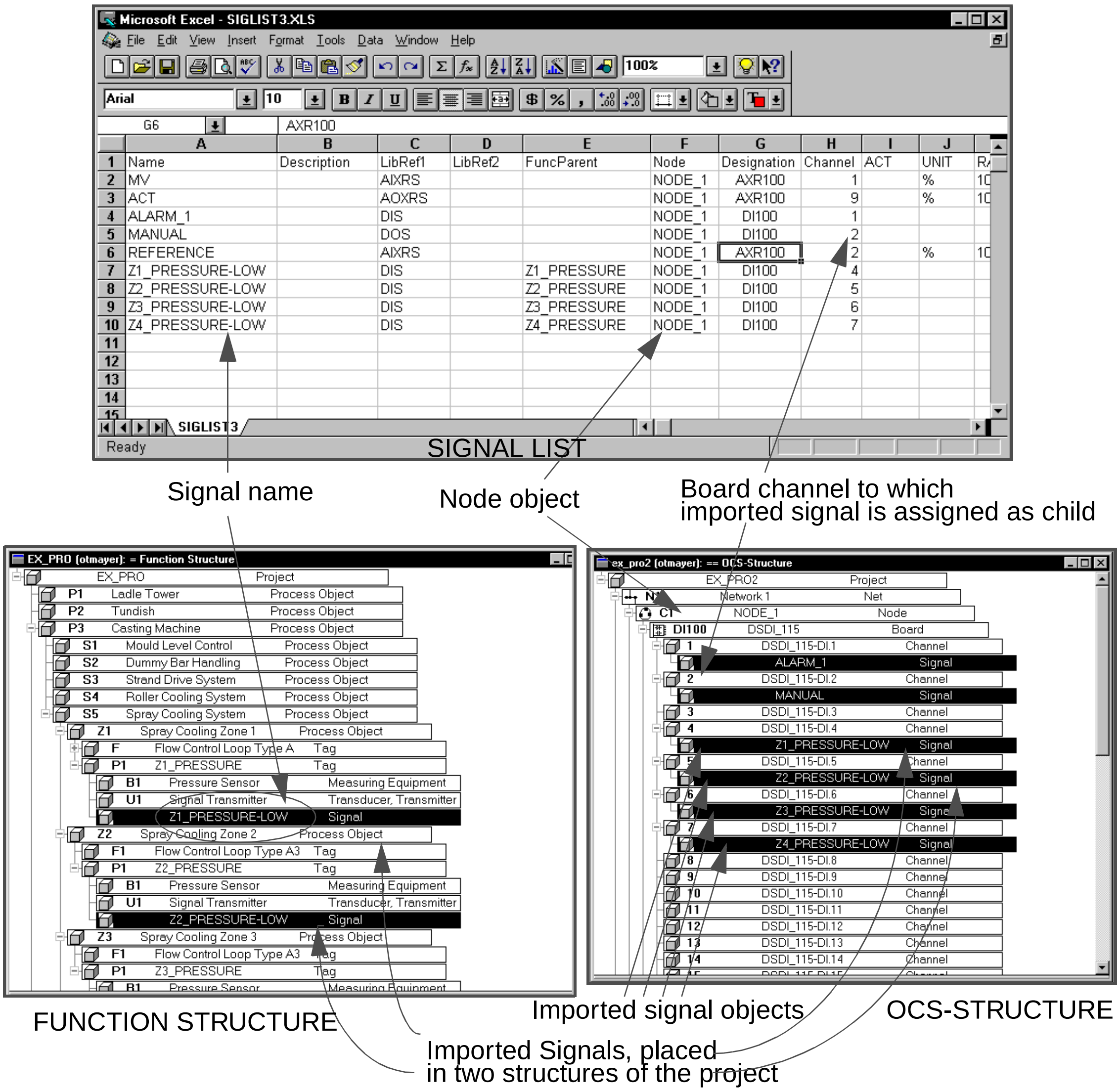
Figure S10-1. Import Dialog

Answer the dialog boxes during the import procedure and check the logfile after the import.

S10.2 Import Results

S10.2.1 Creation of Signals in the Target Project

The items of the signal list are placed by the import procedure into the OCS-Structure of the project. If a reference to an already existing channel is given, the instance is placed as child of this channel. Signals without a channel reference is put as root objects into the OCS-Structure and must be manually assigned to channels. If the name of a signal in the signal list corresponds in a defined way to a name of an object in the Function Structure of the project, the signal instance will also be placed as child to this object in the project Function Structure. In our example we show the creation and placing of signals in spray cooling system via an import function.


Microsoft Excel - SIGLIST3.XLS
File Edit View Insert Format Tools Data Window Help
100%
Arial
10
G6
AXR100
SIGLIST3
Ready
SIGNAL LIST
Signal name
Node object
Board channel to which
imported signal is assigned as child
EX_PRO (otmayer): = Function Structure
ex_pro2 (otmayer): == OCS-Structure
FUNCTION STRUCTURE
Imported signal objects
OCS-STRUCTURE
Imported Signals, placed
in two structures of the project

	A	B	C	D	E	F	G	H	I	J
1	Name	Description	LibRef1	LibRef2	FuncParent	Node	Designation	Channel	ACT	UNIT
2	MV		AIXRS			NODE_1	AXR100	1		%
3	ACT		AOXRS			NODE_1	AXR100	9		%
4	ALARM_1		DIS			NODE_1	DI100	1		
5	MANUAL		DOS			NODE_1	DI100	2		
6	REFERENCE		AIXRS			NODE_1	AXR100	2		%
7	Z1_PRESSURE-LOW		DIS		Z1_PRESSURE	NODE_1	DI100	4		
8	Z2_PRESSURE-LOW		DIS		Z2_PRESSURE	NODE_1	DI100	5		
9	Z3_PRESSURE-LOW		DIS		Z3_PRESSURE	NODE_1	DI100	6		
10	Z4_PRESSURE-LOW		DIS		Z4_PRESSURE	NODE_1	DI100	7		
11										
12										
13										
14										

	EX_PRO	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
P3	Casting Machine	Process Object
S1	Mould Level Control	Process Object
S2	Dummy Bar Handling	Process Object
S3	Strand Drive System	Process Object
S4	Roller Cooling System	Process Object
S5	Spray Cooling System	Process Object
Z1	Spray Cooling Zone 1	Process Object
F	Flow Control Loop Type A	Tag
P1	Z1_PRESSURE	Tag
B1	Pressure Sensor	Measuring Equipment
U1	Signal Transmitter	Transducer, Transmitter
	Z1_PRESSURE-LOW	Signal
Z2	Spray Cooling Zone 2	Process Object
F1	Flow Control Loop Type A3	Tag
P1	Z2_PRESSURE	Tag
B1	Pressure Sensor	Measuring Equipment
U1	Signal Transmitter	Transducer, Transmitter
	Z2_PRESSURE-LOW	Signal
Z3	Spray Cooling Zone 3	Process Object
F1	Flow Control Loop Type A3	Tag
P1	Z3_PRESSURE	Tag
B1	Pressure Sensor	Measuring Equipment

	EX_PRO2	Project
N1	Network 1	Net
C1	NODE_1	Node
DI100	DSDI_115	Board
1	DSDI_115-DI.1	Channel
	ALARM_1	Signal
2	DSDI_115-DI.2	Channel
	MANUAL	Signal
3	DSDI_115-DI.3	Channel
4	DSDI_115-DI.4	Channel
	Z1_PRESSURE-LOW	Signal
5	DSDI_115-DI.5	Channel
	Z2_PRESSURE-LOW	Signal
6	DSDI_115-DI.6	Channel
	Z3_PRESSURE-LOW	Signal
7	DSDI_115-DI.7	Channel
	Z4_PRESSURE-LOW	Signal
8	DSDI_115-DI.8	Channel
9	DSDI_115-DI.9	Channel
10	DSDI_115-DI.10	Channel
11	DSDI_115-DI.11	Channel
12	DSDI_115-DI.12	Channel
13	DSDI_115-DI.13	Channel
14	DSDI_115-DI.14	Channel

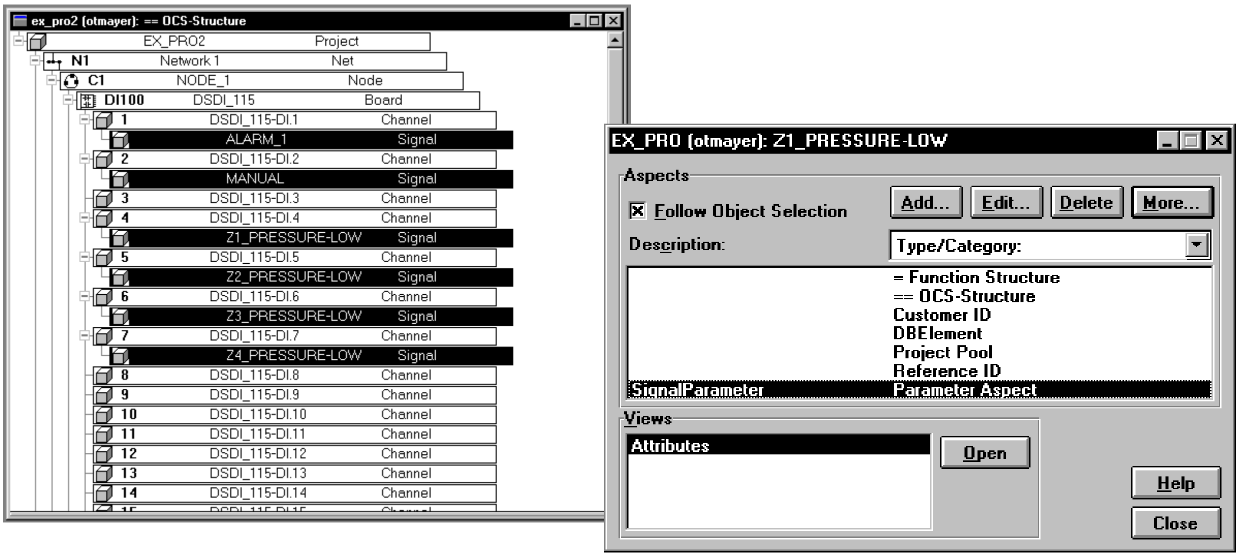
Figure S10-2. Import of Signal Objects

S10.2.2 Aspects and Parameters of Signal Objects

An instance of a signal object gets all the aspects of the corresponding object in the Object library. The parameters in the signal list are put into the Signal Parameter aspect of the signal object instance.

Figure S10-3 shows the aspects of an imported signal object. Figure S10-4 shows the view on the attributes of a Signal Parameter aspect. The parameters are taken over from the signal list but can be edited afterwards by using this view.

The contents of the Signal Parameter aspects are used for creating the DB-elements.



ex_pro2 (otmayer): == OCS-Structure

	EX_PRO2	Project
N1	Network 1	Net
C1	NODE_1	Node
DI100	DSDI_115	Board
1	DSDI_115-DI.1	Channel
	ALARM_1	Signal
2	DSDI_115-DI.2	Channel
	MANUAL	Signal
3	DSDI_115-DI.3	Channel
4	DSDI_115-DI.4	Channel
	Z1_PRESSURE-LOW	Signal
5	DSDI_115-DI.5	Channel
	Z2_PRESSURE-LOW	Signal
6	DSDI_115-DI.6	Channel
	Z3_PRESSURE-LOW	Signal
7	DSDI_115-DI.7	Channel
	Z4_PRESSURE-LOW	Signal
8	DSDI_115-DI.8	Channel
9	DSDI_115-DI.9	Channel
10	DSDI_115-DI.10	Channel
11	DSDI_115-DI.11	Channel
12	DSDI_115-DI.12	Channel
13	DSDI_115-DI.13	Channel
14	DSDI_115-DI.14	Channel

EX_PRO (otmayer): Z1_PRESSURE-LOW

Aspects

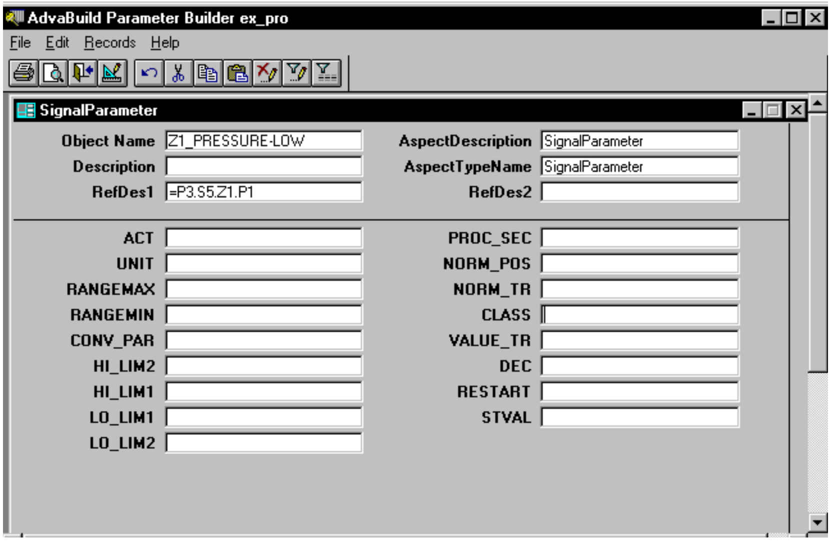
☒ Follow Object Selection

Description:	Type/Category:
	= Function Structure
	== OCS-Structure
	Customer ID
	DBElement
	Project Pool
	Reference ID
SignalParameter	Parameter Aspect

Views

Attributes

Figure S10-3. Imported Signal Objects



AdvaBuild Parameter Builder ex_pro

SignalParameter

Object Name	Z1_PRESSURE-LOW	AspectDescription	SignalParameter
Description		AspectTypeName	SignalParameter
RefDes1	=P3.S5.Z1.P1	RefDes2	
ACT		PROC_SEC	
UNIT		NORM_POS	
RANGEMAX		NORM_TR	
RANGEMIN		CLASS	
CONV_PAR		VALUE_TR	
HI_LIM2		DEC	
HI_LIM1		RESTART	
LO_LIM1		STVAL	
LO_LIM2			

Figure S10-4. View on the Attributes of a Signal Parameter Aspect

S10.2.3 Reimport of Signal Data

You can perform a reimport of signal data from a signal data list that has already been imported. This is usually done after modifications or further completion of the data in the signal list. You can perform reimport either in the keep mode or in the overwrite mode. In the keep mode an already existing aspect of a signal object is kept with its former data. In the overwrite mode an already existing aspect of a signal object is deleted and replaced by the data from the signal list used for reimport.

NOTE

The identity of an aspect is evaluated from the aspect type and the aspect description. If you change an aspect description within a signal object and you perform a reimport with the old aspect description still in the signal list, this aspect will be handled like an additional aspect in the import result. Therefore aspect descriptions must not be changed if a reimport is performed.

Step 11 Build PC-Program

S11.1 Create Node Control Software Aspect

PC-programs are assigned to Node Control Software aspects of Node objects. Therefore, first create this aspect.

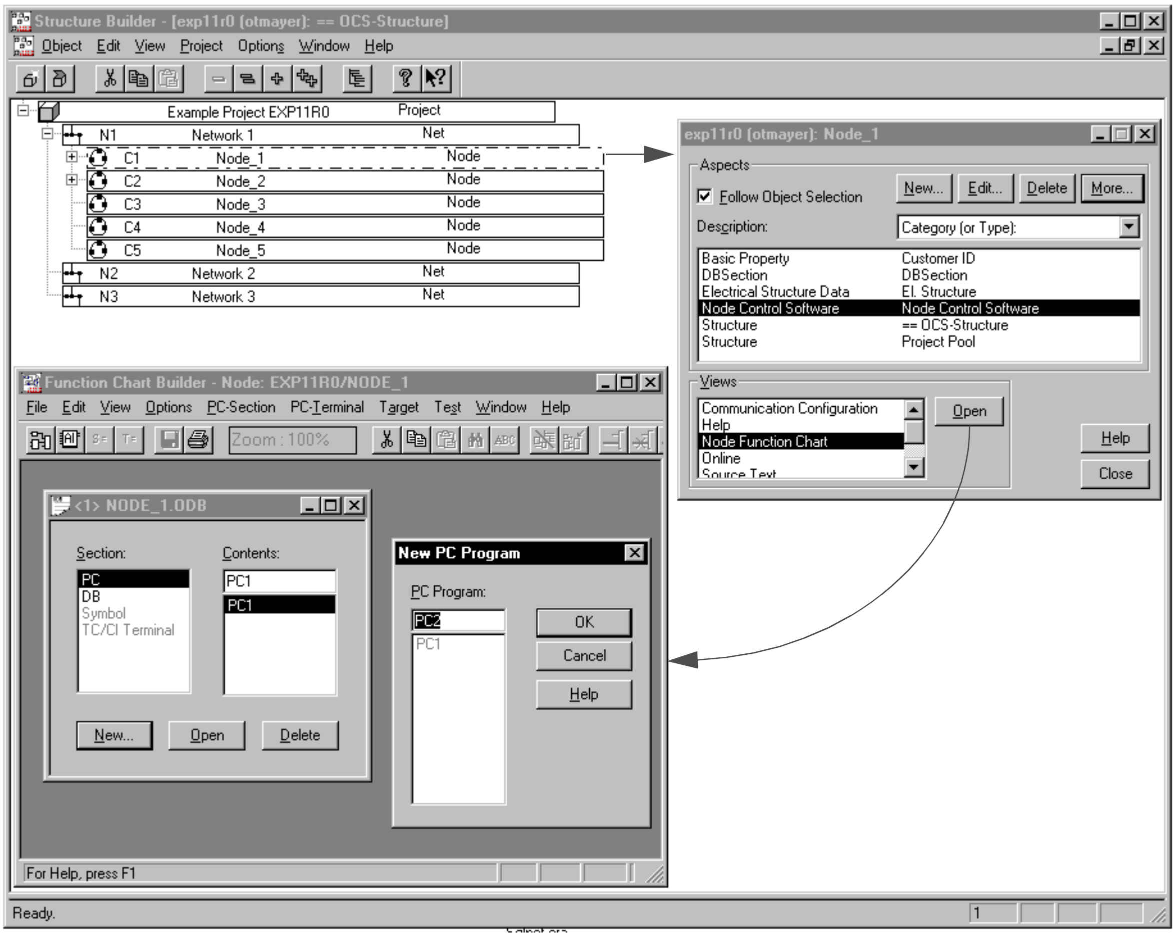
Creation of a Node Control Software Aspect	
Preconditions:	Node object created
User activities:	1. Use the Structure Builder 2. Open the object window of the node object to which PC-program(s) shall be assigned and click on **New** 3. Select **Node Control Software** in the **New Aspect** window and fill in the Aspect Description 4. Click on **OK**
Result:	Node Control Software aspect created

S11.2 Create PC-Program in Node Control Software Aspect

Assign PC-programs to Node Control Software aspects via the Function Chart Builder.

Creation of a PC-Program	
Preconditions:	Node object created and Node Control Software aspect assigned
User activities:	1. Use the Structure Builder 2. Open the **Node Function Chart** view of the Node Control Software aspect 3. Create the relevant new sections for PC, DB, Symbols and Terminals 4. Edit the relevant new sections for PC, DB, Symbols and Terminals 5. Click on **OK**
Result:	PC-program created

Figure S11-1 shows the Object window of a node object and the Node Function Chart view on the Node Control Software aspect. The PC-program PC1 is inserted in the shown example.


Structure Builder - [exp11r0 (otmayer): == OCS-Structure]
Object Edit View Project Options Window Help
exp11r0 (otmayer): Node_1
Aspects
Follow Object Selection
New... Edit... Delete More...
Description:
Views
Open
Help
Close
Function Chart Builder - Node: EXP11R0/NODE_1
File Edit View Options PC-Section PC-Terminal Target Test Window Help
Zoom : 100%
<1> NODE_1.ODB
Section:
Contents:
New... Open Delete
New PC Program
PC Program:
OK
Cancel
Help
For Help, press F1
Ready.
1

	Example Project EXP11R0	Project
N1	Network 1	Net
C1	Node_1	Node
C2	Node_2	Node
C3	Node_3	Node
C4	Node_4	Node
C5	Node_5	Node
N2	Network 2	Net
N3	Network 3	Net

☑ Follow Object Selection

Description:	Category (or Type):
Basic Property	Customer ID
DBSection	DBSection
Electrical Structure Data	El. Structure
Node Control Software	Node Control Software
Structure	== OCS-Structure
Structure	Project Pool

Views:
Communication Configuration
Help
Node Function Chart
Online
Source Text

Section:
PC
DB
Symbol
TC/CI Terminal

Contents:
PC1
PC1

PC Program: PC2
PC1

Figure S11-1. Inserting a PC-Program in a Node Control Software Aspect

Step 12 Perform I/O Allocation

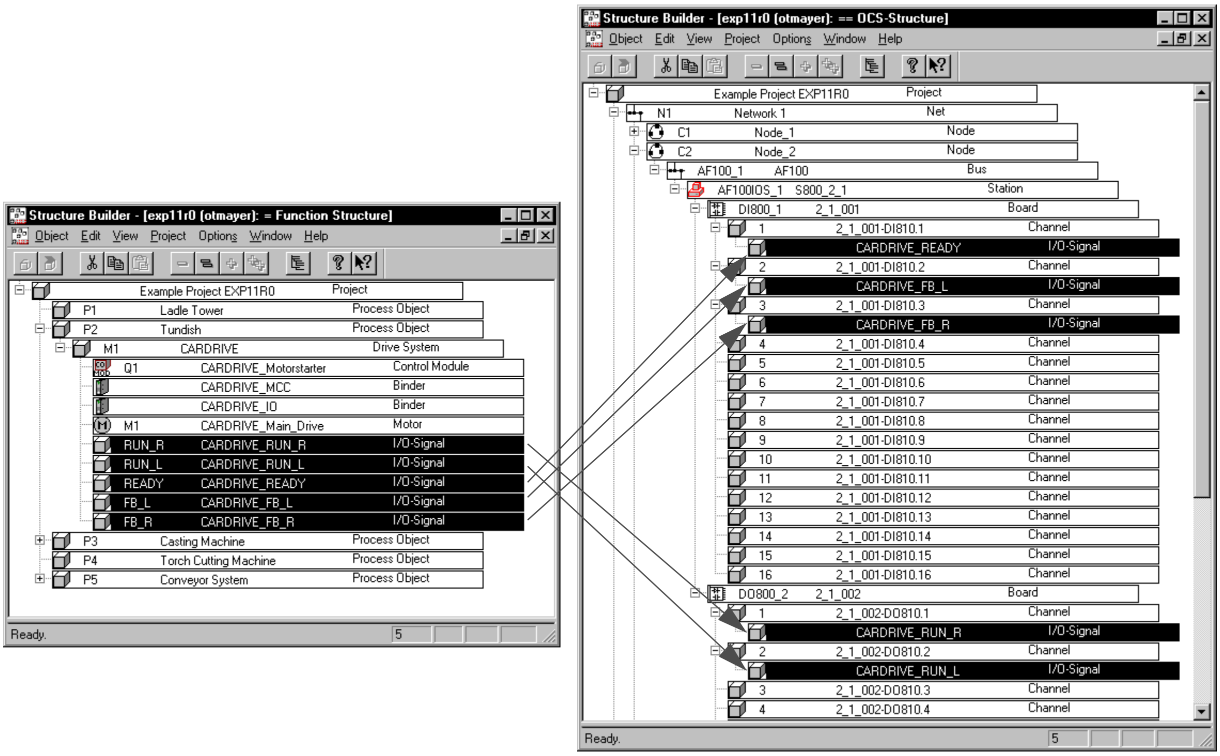
S12.1 Drag and Drop Signals to Channels

The I/O allocation covers the allocation of signals to channels of OCS-boards. Signals are modeled as Advant objects. They are usually placed in the Function Structure and in the Control System Structure of the target project. All data relevant for a signal is assigned to aspects of the Advant object representing the signal.

The mapping of signals to I/O channels of OCS-boards is performed by placing signals as children objects to channels. You can do this efficiently by, for example, using the drag and drop functionality for the placing of objects. Allocate bulk signals by using the import functionality of the AdvaBuild Engineering Software as described in Step 10.

S12.2 Placing Signals of the Drive System Example

Assign the I/O signals of a drive system example to S800 boards in an S800 I/O station. Figure shows this assignment.


Structure Builder - [exp11r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
Ready.
5
Structure Builder - [exp11r0 (otmayer): == OCS-Structure]
Object Edit View Project Options Window Help
Ready.
5

	Example Project EXP11R0	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
M1	CARDRIVE	Drive System
Q1	CARDRIVE_Motorstarter	Control Module
	CARDRIVE_MCC	Binder
	CARDRIVE_IO	Binder
M1	CARDRIVE_Main_Drive	Motor
RUN_R	CARDRIVE_RUN_R	I/O-Signal
RUN_L	CARDRIVE_RUN_L	I/O-Signal
READY	CARDRIVE_READY	I/O-Signal
FB_L	CARDRIVE_FB_L	I/O-Signal
FB_R	CARDRIVE_FB_R	I/O-Signal
P3	Casting Machine	Process Object
P4	Torch Cutting Machine	Process Object
P5	Conveyor System	Process Object

	Example Project EXP11R0	Project
N1	Network 1	Net
C1	Node_1	Node
C2	Node_2	Node
AF100_1	AF100	Bus
AF100IOS_1	S800_2_1	Station
DI800_1	2_1_001	Board
1	2_1_001-DI810.1	Channel
	CARDRIVE_READY	I/O-Signal
2	2_1_001-DI810.2	Channel
	CARDRIVE_FB_L	I/O-Signal
3	2_1_001-DI810.3	Channel
	CARDRIVE_FB_R	I/O-Signal
4	2_1_001-DI810.4	Channel
5	2_1_001-DI810.5	Channel
6	2_1_001-DI810.6	Channel
7	2_1_001-DI810.7	Channel
8	2_1_001-DI810.8	Channel
9	2_1_001-DI810.9	Channel
10	2_1_001-DI810.10	Channel
11	2_1_001-DI810.11	Channel
12	2_1_001-DI810.12	Channel
13	2_1_001-DI810.13	Channel
14	2_1_001-DI810.14	Channel
15	2_1_001-DI810.15	Channel
16	2_1_001-DI810.16	Channel
DO800_2	2_1_002	Board
1	2_1_002-DO810.1	Channel
	CARDRIVE_RUN_R	I/O-Signal
2	2_1_002-DO810.2	Channel
	CARDRIVE_RUN_L	I/O-Signal
3	2_1_002-DO810.3	Channel
4	2_1_002-DO810.4	Channel

Figure S12-1. Placing Signals

Step 13 Build/Adapt Modules

S13.1 Create Circuit Aspect

A control module is assigned to a Circuit aspect of an Advant object. First create the Circuit aspect.

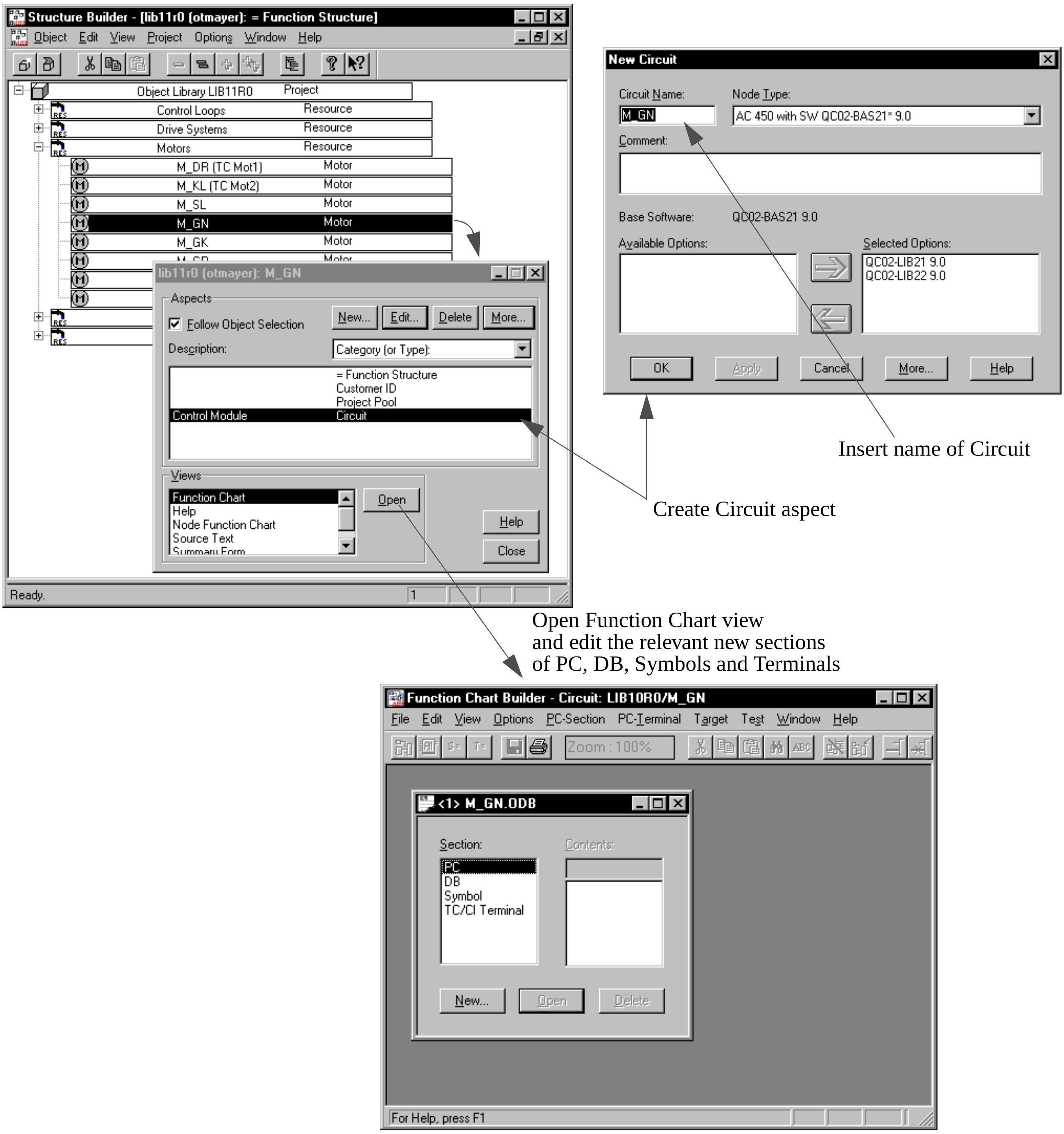
Creation of a Circuit Aspect	
Preconditions:	Object created
User activities:	1. Use the Structure Builder 2. Open the object window of the object to which a Circuit shall be assigned and click on **New** 3. Select **Circuit** in the **New Aspect** window and fill in the Aspect Description 4. Click on **OK**
Result:	Circuit aspect created

S13.2 Create Control Module in Circuit Aspect

Create control modules in circuit aspects by the Function Chart Builder.

Creation of a Control Module	
Preconditions:	Object created and Circuit aspect assigned
User activities:	1. Use the Structure Builder 2. Open the **Function Chart** view of the **Circuit** aspect 3. Create the relevant new sections for PC, DB, Symbols and Terminals 4. Edit the relevant sections of PC, DB, Symbols and Terminals
Result:	Control Module created

Figure S13-1 shows the windows involved when a Control Module is created.



Structure Builder - [lib11r0 (otmayer): = Function Structure]

Object Library LIB11R0	Project
Control Loops	Resource
Drive Systems	Resource
Motors	Resource
M_DR (TC Mot1)	Motor
M_KL (TC Mot2)	Motor
M_SL	Motor
M_GN	Motor
M_GK	Motor

lib11r0 (otmayer): M_GN

☑ Follow Object Selection

Description: Category (or Type):

	= Function Structure
	Customer ID
	Project Pool
Control Module	Circuit

Views: Function Chart, Help, Node Function Chart, Source Text, Summary Form

New Circuit

Circuit Name: M_GN

Node Type: AC 450 with SW QC02-BAS21* 9.0

Comment:

Base Software: QC02-BAS21 9.0

Available Options:

Selected Options: QC02-LIB21 9.0, QC02-LIB22 9.0

Insert name of Circuit

Create Circuit aspect

Open Function Chart view
and edit the relevant new sections
of PC, DB, Symbols and Terminals

Function Chart Builder - Circuit: LIB10R0/M_GN

<1> M_GN.ODB

Section: PC, DB, Symbol, TC/CI Terminal

Contents:

Figure S13-1. Inserting a Control Module into a Circuit Aspect

Step 14 Instantiate SW Modules

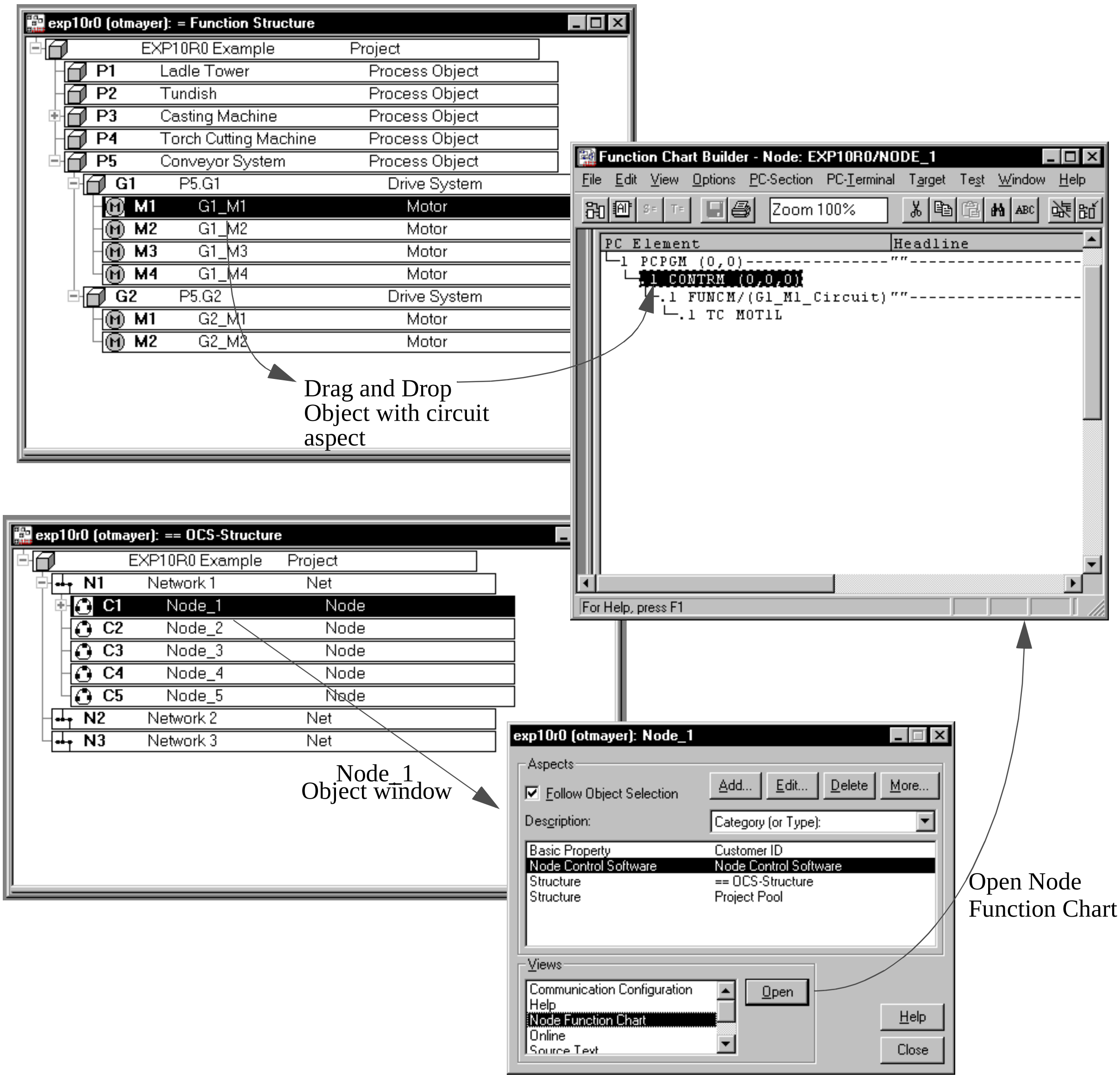
S14.1 Using Circuits in the Node Mode

Use circuits by instantiating them into PC-programs in Node Mode. Instantiation is done by using drag and drop. The object to drag and drop is the Advant object having one ore more circuit aspects. It can be dropped on PC elements shown in a PC program function chart window or tree window.

Placing Circuits in a Node Function Chart	
Preconditions:	Node created and Circuits as aspects of Advant objects created
User activities:	1. Use the Structure Builder 2. Select Node 3. Select **Node Control Software** aspect 4. Open **Node Function Chart** view 5. Create or Open **PC Program** 6. Select the Advant object that contains in a circuit aspect the circuit to be placed 7. Drag the Advant object and drop it on the PC element in the Function Chart where the circuit shall be placed. Pressing **Shift** during drag and drop inserts the circuit on same level, not pressing **Shift** on lower level.
Result:	Circuits placed and expanded in the Node Function Chart

The item designation of the starting structure element of an inserted circuit (typically FUNCM) gets the drop item designation and an instance name composed of the customer ID of the Advant object concatenated with the aspect description of the circuit aspect.

Figure S14-1 presents the view on the Function Structure containing the object with the circuit that shall be instantiated and the Node Function Chart view of the PC-program to which the circuit is instantiated by using drag and drop.



	EXP10R0 Example	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
P3	Casting Machine	Process Object
P4	Torch Cutting Machine	Process Object
P5	Conveyor System	Process Object
G1	P5.G1	Drive System
M1	G1_M1	Motor
M2	G1_M2	Motor
M3	G1_M3	Motor
M4	G1_M4	Motor
G2	P5.G2	Drive System
M1	G2_M1	Motor
M2	G2_M2	Motor

```
PC Element                                Headline
└1 PCPGM (0,0)---------------""----------------
  └.1 CONTRM (0,0,0)
     .1 FUNCM/(G1_M1_Circuit)""----------------
      └.1 TC MOT1L
```

	EXP10R0 Example	Project
N1	Network 1	Net
C1	Node_1	Node
C2	Node_2	Node
C3	Node_3	Node
C4	Node_4	Node
C5	Node_5	Node
N2	Network 2	Net
N3	Network 3	Net

☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
Node Control Software	Node Control Software
Structure	== OCS-Structure
Structure	Project Pool

Communication Configuration
Help
Node Function Chart
Online
Source Text

Drag and Drop
Object with circuit
aspect

Node_1
Object window

Open Node
Function Chart

Figure S14-1. Instantiating the Circuit Aspect of an Object by Drag and Drop

Step 15 Interconnect SW Modules

S15.1 Create Glue Logic

After instantiating the circuits into the PC-programs you can design the additional glue logic to connect the modules and to complete the PC-programs.

Perform this by opening the Node Function Chart view of the Node Control Software aspect of the relevant node. Create the glue logic by editing the Node Function Chart.

Open Node Function Chart	
User activities:	1. Use the Structure Builder 2. Open the ***Object*** window of the relevant node 3. Open the **Node Function Chart** view of the **Node Control Software** aspect 4. Choose **Open Section | PC** from the **File** menu of the FCB 5. Select the PC program you want to edit from the **Open PC Program** box 6. Click on **OK**
Result:	Node Function Chart open for inspection and editing

Figure S15-1 shows the Node Function Chart view on a PC Program while editing glue logic.

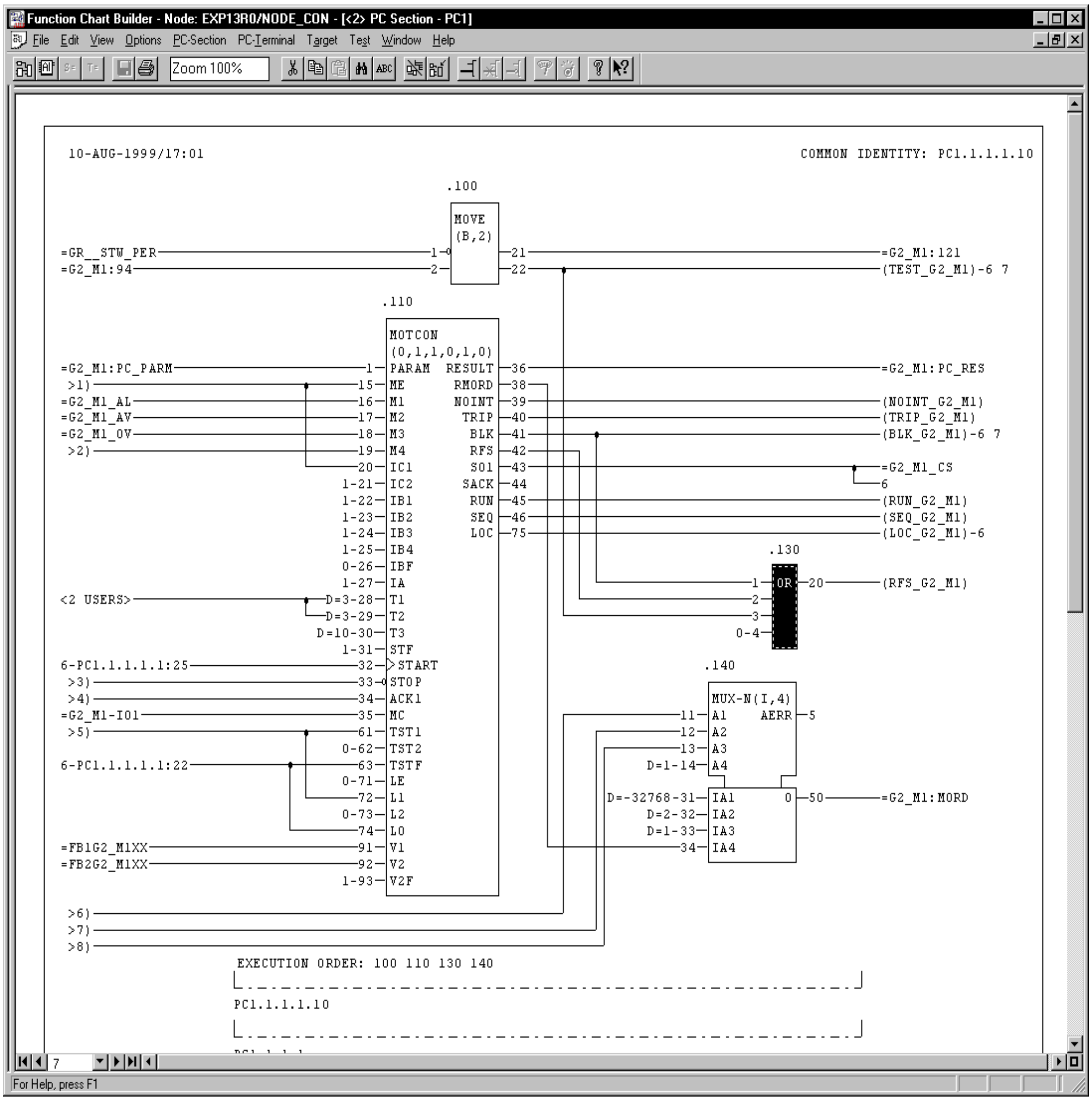


Figure S15-1. Editing a Node Function Chart

Step 16 Perform Parameter Update

S16.1 Summary of Parameter Update Procedures

Parameter update is the process of setting symbols defined in Type Circuits and Circuits to the values specified in parameter aspects. You can perform the parameter update for:

- single objects containing circuit aspects and corresponding parameter aspects
- all objects in the project containing circuits with corresponding parameter aspects.

You can perform the parameter update for an object on the:

- Function Chart of a Circuit
- Node Function Chart of the instantiated Circuit.

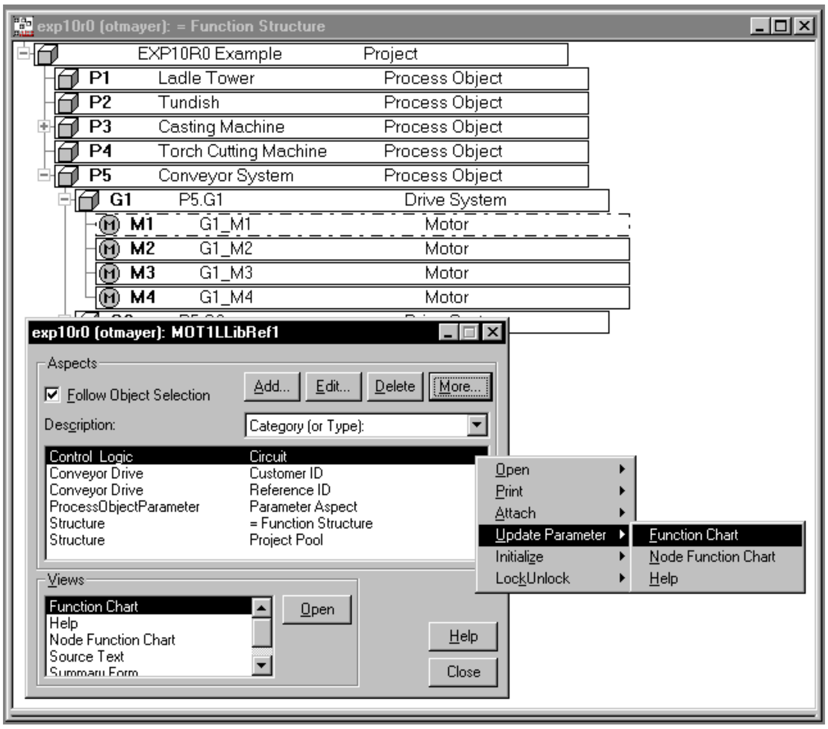
NOTE

Updating parameters in a circuit means updating of symbol values defined in type circuits that are instantiated in the circuit.

S16.2 Updating Parameters in the Function Chart of a Circuit

Updating Parameters of a Circuit	
Preconditions:	Circuit and corresponding Parameter aspect created
User activities:	1. Use the Structure Builder 2. Open ***Object*** window and click on the **Circuit** aspect 3. Click on **More** in the ***Object*** window 4. Click on **Update Parameter | Function Chart**
Result:	Symbol values Circuits updated

Figure S16-1 illustrates the parameter update of a circuit.



	EXP10R0 Example	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
P3	Casting Machine	Process Object
P4	Torch Cutting Machine	Process Object
P5	Conveyor System	Process Object
G1	P5.G1	Drive System
M1	G1_M1	Motor
M2	G1_M2	Motor
M3	G1_M3	Motor
M4	G1_M4	Motor

☑ Follow Object Selection

Description: Category (or Type):

Control Logic	Circuit
Conveyor Drive	Customer ID
Conveyor Drive	Reference ID
ProcessObjectParameter	Parameter Aspect
Structure	= Function Structure
Structure	Project Pool

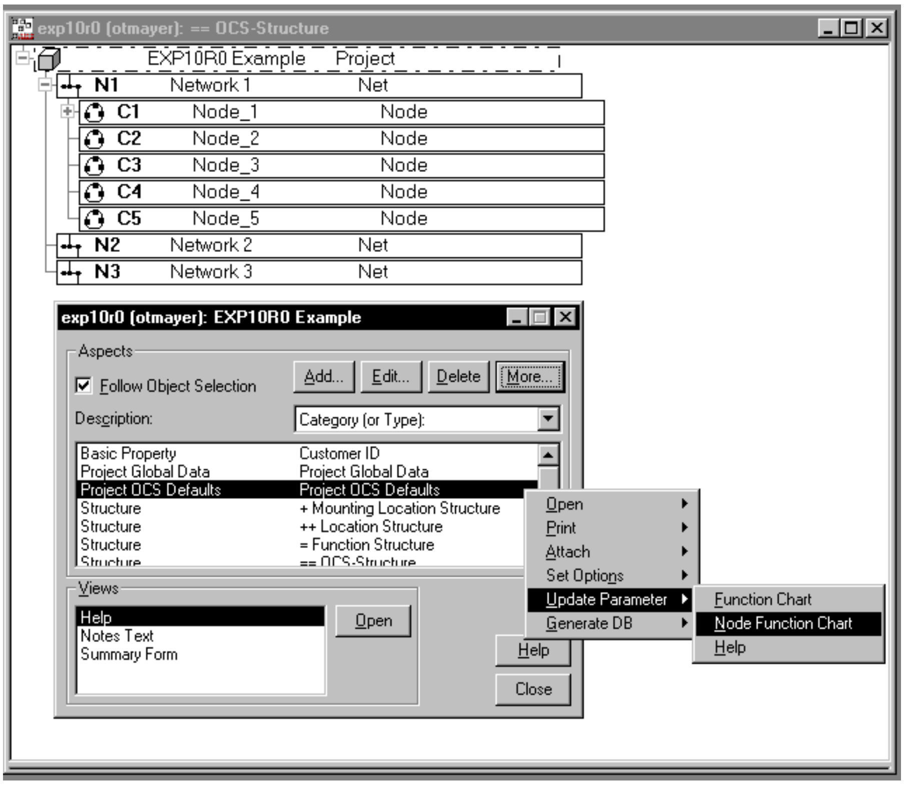
Views: Function Chart, Help, Node Function Chart, Source Text, Summary Form

Figure S16-1. Updating the Parameters of a Circuit

S16.3 Updating the Parameters for a Complete Project

Updating Parameters for a Complete Project	
Preconditions:	Node and Circuits created
User activities:	1. Use the Structure Builder 2. Open the **Project** object window and click on the **Project OCS Defaults** aspect 3. Click on **More** in the **Project** object window 4. Click on **Update Parameter | Node Function Chart**
Result:	Symbol values of all Circuits in Node Function Charts updated

Figure S16-2 shows the parameter update on the Project OCS Defaults for a complete project.



	EXP10R0 Example	Project
N1	Network 1	Net
C1	Node_1	Node
C2	Node_2	Node
C3	Node_3	Node
C4	Node_4	Node
C5	Node_5	Node
N2	Network 2	Net
N3	Network 3	Net

☑ Follow Object Selection

Description:	Category (or Type):
Basic Property	Customer ID
Project Global Data	Project Global Data
Project OCS Defaults	Project OCS Defaults
Structure	+ Mounting Location Structure
Structure	++ Location Structure
Structure	= Function Structure
Structure	== OCS-Structure

Views:
Help
Notes Text
Summary Form

Figure S16-2. Updating the Parameters for a Complete Project

Step 17 Generate DB Elements

S17.1 Generate DB Elements for Node Control Software

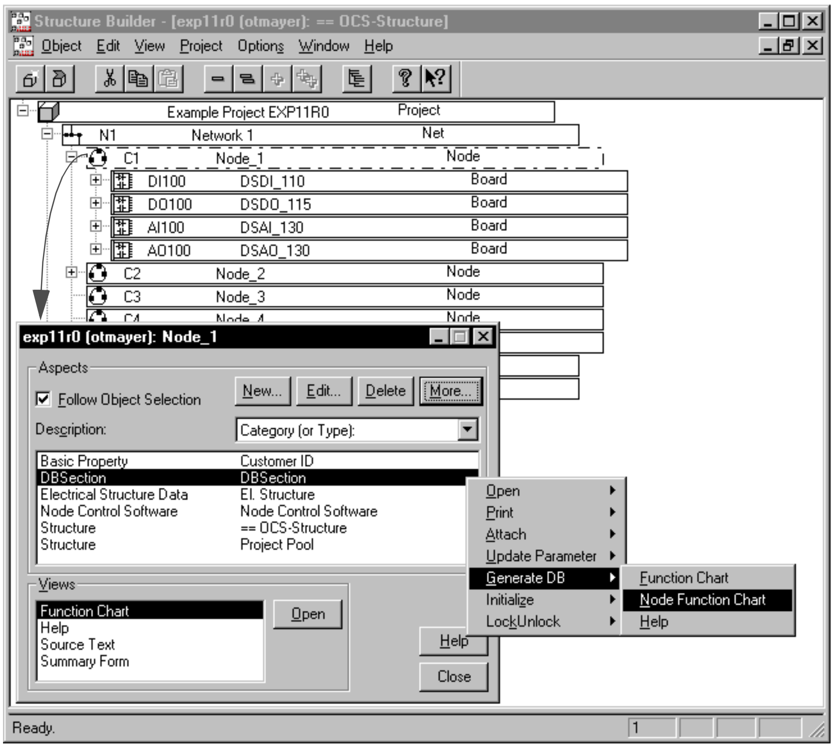
Board and signal DB-elements are represented in the board and signal objects by the aspect type DB Element.

All board and signal DB elements in a subtree are represented by an aspect of the type DB Section.

Data of the DB Section aspect is accumulated by the data of the DB Element aspects in the OCS-Structure below the Advant object carrying the DB section aspect. Suitable Advant objects for carrying the DB section aspect are, for example, the node objects or objects directly below node objects. DB elements are generated by the following steps:

Generate DB Elements	
Preconditions:	DB section aspect in or below node object created
User activities:	1. Use the Structure Builder 2. Open Object window and click on the **DB Section** aspect 3. Click on **More** in the Object window 4. Click on **Generate DB | Node Function Chart**
Result:	DB Elements generated in the Node Control Software

Figure S17-1 illustrates the generate DB procedure.



	Example Project EXP11R0	Project
N1	Network 1	Net
C1	Node_1	Node
DI100	DSDI_110	Board
DO100	DSDO_115	Board
AI100	DSAI_130	Board
AO100	DSAO_130	Board
C2	Node_2	Node
C3	Node_3	Node
C4	Node_4	Node

Basic Property	Customer ID
DBSection	DBSection
Electrical Structure Data	El. Structure
Node Control Software	Node Control Software
Structure	== OCS-Structure
Structure	Project Pool

Figure S17-1. Generation of all DB Elements of a Node

Step 18 Build Mounting Locations

S18.1 Engineering Related to Mounting Locations

When continuing the plant engineering, make more detailed estimations about the types and amounts of devices and derive from that estimations the types and amounts of cabinets needed to place the devices. Organize the cabinets used in the plant in a Mounting Location Structure.

The mounting locations are the units on/in which the devices of the electrical and control system are mounted. Examples of such mountings are:

- Boards in racks
- Racks in cabinets
- Terminal units in cabinets
- Motor controllers in motor controller cabinets
- Connection units on cabinet subunits
- Transmitters in protection boxes.

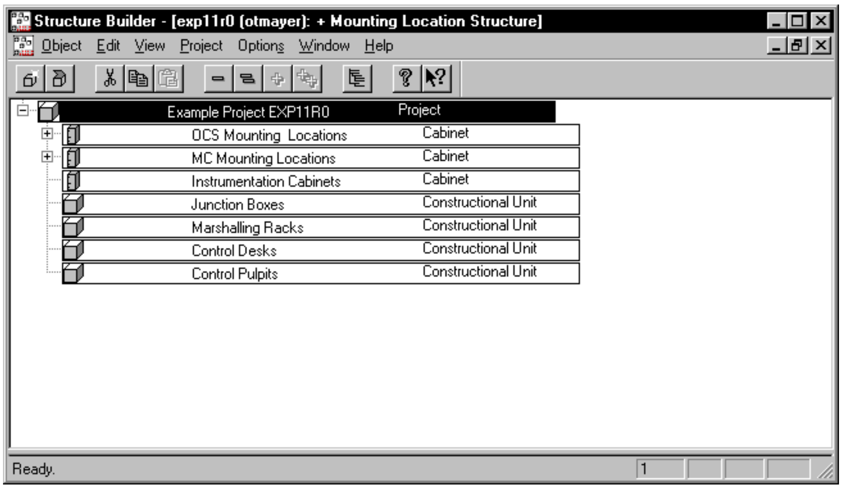
The items representing mounting locations are placed in later planning steps in the Topographic Location Structure. By doing this, continuous navigation, paths are created, leading from the project object via the Topographic Location Structure and the Mounting Location Structure down to the devices in cabinets, desks, and other possible constructional units carrying devices.

S18.1.1 Building the Mounting Location Structure

The Mounting Location Structure is an auxiliary structure which makes it possible to engineer complete constructional units, like cabinets and desks, without exactly knowing where they will be placed. For example, it is possible to do the detail engineering for a cabinet in a first step and locate this cabinet in the Topographic Location Structure in a second step.

The project object is the top object if the Mounting Location Structure comprises all mounting locations in the plant. In the level below, we arrange the objects representing typical mounting location units (constructional units that can carry devices) like:

- Control system cabinets
- Motor controller cabinets
- Instrumentation cabinets
- Marshalling racks
- Junction boxes
- Operator desks
- Pulpits.


Structure Builder - [exp11r0 (otmayer): + Mounting Location Structure]
Object Edit View Project Options Window Help
Ready.
1

Example Project EXP11R0	Project
OCS Mounting Locations	Cabinet
MC Mounting Locations	Cabinet
Instrumentation Cabinets	Cabinet
Junction Boxes	Constructional Unit
Marshalling Racks	Constructional Unit
Control Desks	Constructional Unit
Control Pulpits	Constructional Unit

Figure S18-1. Higher Levels of a Mounting Location Structure

Figure S18-1 is an example of the higher levels of a Mounting Location Structure. The objects on this level represent classes of mounting locations (constructional units). You are free to build up this structure, and all the other structures, the way you want them.

S18.1.2 Working with Standard Solutions of Mounting Locations

Standard solutions are available for most constructional units used as mounting locations. Such standard solutions define the constructional unit, for example, a control cabinet with its racks and slots. Central processing devices, power supply devices and terminal bars may already be defined and located in the standard solution of such a cabinet. By using such standards the engineering work is reduced to selection, placing and wiring of the I/O-boards.

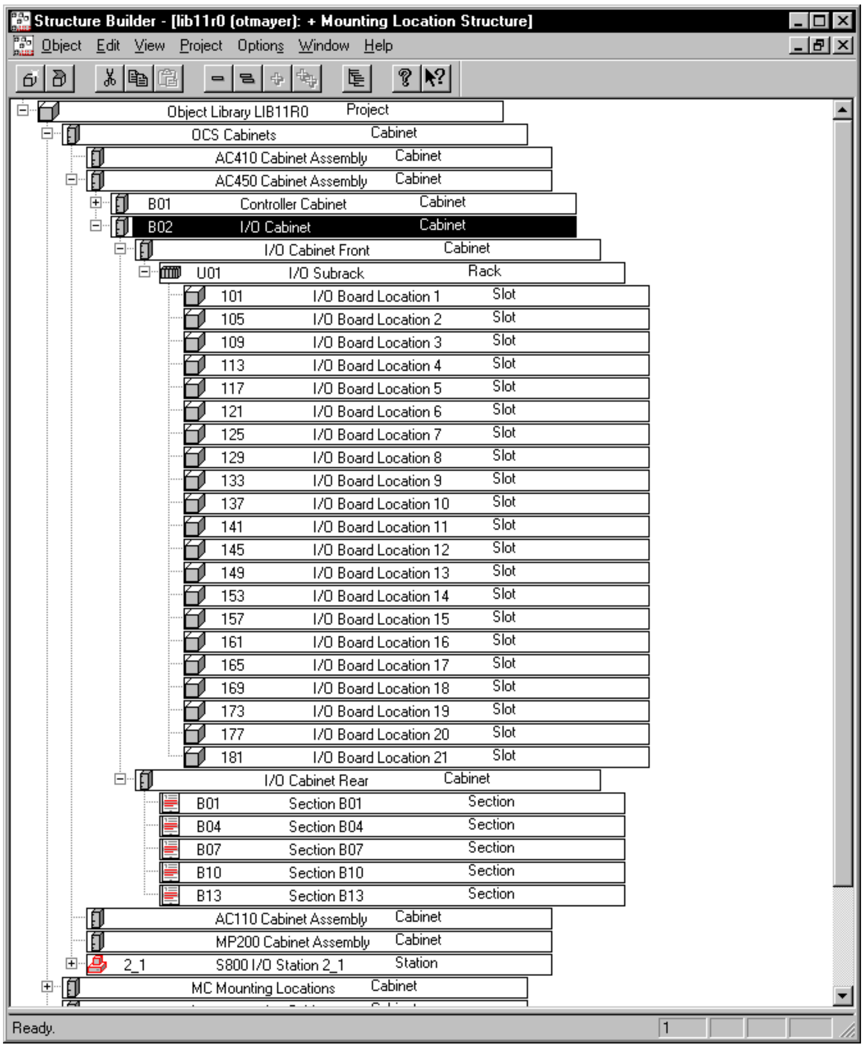
The standard solutions to use in the project are placed in the Object library.

The higher levels of the Mounting Location Structure in the Object library should be the same as in the example project shown in Figure S18-1.

Creating an AC 450 Cabinet Assembly

Configure a standard control system cabinet and put it into the Object library.

Figure S18-2 shows an example of a standard solution for an OCS cabinet assembly in the Object library. The cabinet assembly consists of a controller cabinet and an I/O cabinet with an I/O subrack with its slots for placing boards. Placing the boards is done in the detail engineering. The controller subrack is not structured into further details since it is a standard arrangement with all boards included and no further engineering necessary.


Structure Builder - [lib11r0 (otmayer): + Mounting Location Structure]
Object
Edit
View
Project
Options
Window
Help
Ready.
1

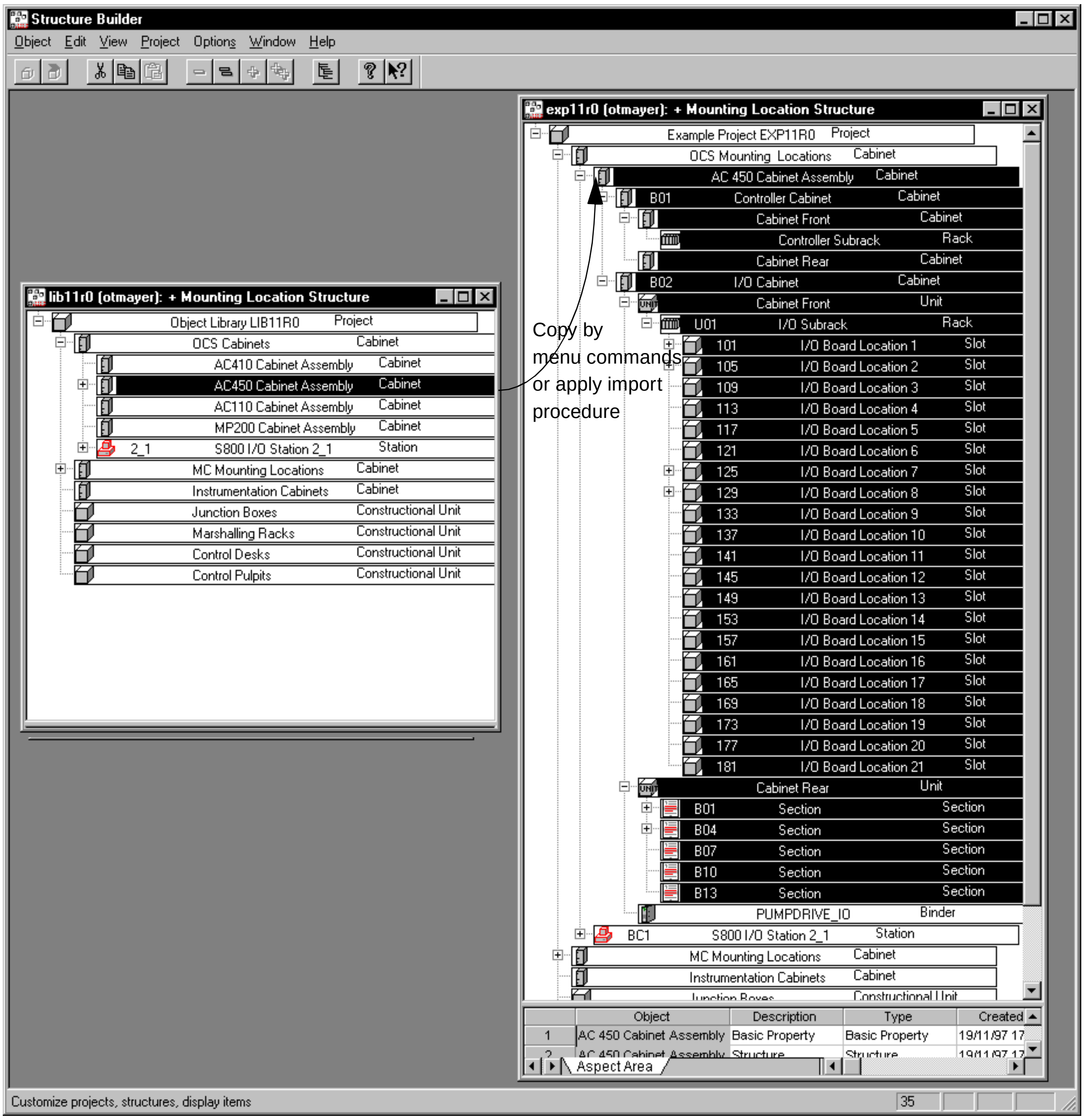
```
Object Library LIB11R0                      Project
  OCS Cabinets                              Cabinet
    AC410 Cabinet Assembly                  Cabinet
    AC450 Cabinet Assembly                  Cabinet
      B01    Controller Cabinet             Cabinet
      B02    I/O Cabinet                    Cabinet
        I/O Cabinet Front                   Cabinet
          U01    I/O Subrack                Rack
            101    I/O Board Location 1     Slot
            105    I/O Board Location 2     Slot
            109    I/O Board Location 3     Slot
            113    I/O Board Location 4     Slot
            117    I/O Board Location 5     Slot
            121    I/O Board Location 6     Slot
            125    I/O Board Location 7     Slot
            129    I/O Board Location 8     Slot
            133    I/O Board Location 9     Slot
            137    I/O Board Location 10    Slot
            141    I/O Board Location 11    Slot
            145    I/O Board Location 12    Slot
            149    I/O Board Location 13    Slot
            153    I/O Board Location 14    Slot
            157    I/O Board Location 15    Slot
            161    I/O Board Location 16    Slot
            165    I/O Board Location 17    Slot
            169    I/O Board Location 18    Slot
            173    I/O Board Location 19    Slot
            177    I/O Board Location 20    Slot
            181    I/O Board Location 21    Slot
        I/O Cabinet Rear                    Cabinet
          B01    Section B01                Section
          B04    Section B04                Section
          B07    Section B07                Section
          B10    Section B10                Section
          B13    Section B13                Section
    AC110 Cabinet Assembly                  Cabinet
    MP200 Cabinet Assembly                  Cabinet
    2_1    S800 I/O Station 2_1             Station
  MC Mounting Locations                     Cabinet
```

Figure S18-2. Structure of a Standard Control System Cabinet

Using Standard Solutions for Mounting Locations by Copying

The cabinet for the controller node of our example, in which the functions for the spray cooling system should be realized, should be a standard solution. This solution is copied from the Object library and placed in the Mounting Location Structure of the project.

Figure S18-3 shows the selected cabinet assembly in the Object library and the copy of this cabinet assembly with all its objects in the Mounting Location Structure window of the example project. Standard solutions should be available for all types of mounting locations used in a project.


Structure Builder
Object Edit View Project Options Window Help
lib11r0 (otmayer): + Mounting Location Structure
exp11r0 (otmayer): + Mounting Location Structure
Aspect Area
35

Copy by menu commands or apply import procedure

	Object Library LIB11R0	Project
	OCS Cabinets	Cabinet
	AC410 Cabinet Assembly	Cabinet
	AC450 Cabinet Assembly	Cabinet
	AC110 Cabinet Assembly	Cabinet
	MP200 Cabinet Assembly	Cabinet
2_1	S800 I/O Station 2_1	Station
	MC Mounting Locations	Cabinet
	Instrumentation Cabinets	Cabinet
	Junction Boxes	Constructional Unit
	Marshalling Racks	Constructional Unit
	Control Desks	Constructional Unit
	Control Pulpits	Constructional Unit

	Example Project EXP11R0	Project
	OCS Mounting Locations	Cabinet
	AC 450 Cabinet Assembly	Cabinet
B01	Controller Cabinet	Cabinet
	Cabinet Front	Cabinet
	Controller Subrack	Rack
	Cabinet Rear	Cabinet
B02	I/O Cabinet	Cabinet
	Cabinet Front	Unit
U01	I/O Subrack	Rack
101	I/O Board Location 1	Slot
105	I/O Board Location 2	Slot
109	I/O Board Location 3	Slot
113	I/O Board Location 4	Slot
117	I/O Board Location 5	Slot
121	I/O Board Location 6	Slot
125	I/O Board Location 7	Slot
129	I/O Board Location 8	Slot
133	I/O Board Location 9	Slot
137	I/O Board Location 10	Slot
141	I/O Board Location 11	Slot
145	I/O Board Location 12	Slot
149	I/O Board Location 13	Slot
153	I/O Board Location 14	Slot
157	I/O Board Location 15	Slot
161	I/O Board Location 16	Slot
165	I/O Board Location 17	Slot
169	I/O Board Location 18	Slot
173	I/O Board Location 19	Slot
177	I/O Board Location 20	Slot
181	I/O Board Location 21	Slot
	Cabinet Rear	Unit
B01	Section	Section
B04	Section	Section
B07	Section	Section
B10	Section	Section
B13	Section	Section
	PUMPDRIVE_IO	Binder
BC1	S800 I/O Station 2_1	Station
	MC Mounting Locations	Cabinet
	Instrumentation Cabinets	Cabinet
	Junction Boxes	Constructional Unit

	Object	Description	Type	Created
1	AC 450 Cabinet Assembly	Basic Property	Basic Property	19/11/97 17
2	AC 450 Cabinet Assembly	Structure	Structure	19/11/97 17

Customize projects, structures, display items

Figure S18-3. Copying a Standard OCS-Cabinet Assembly

Assigning Aspects and Data to Mounting Locations

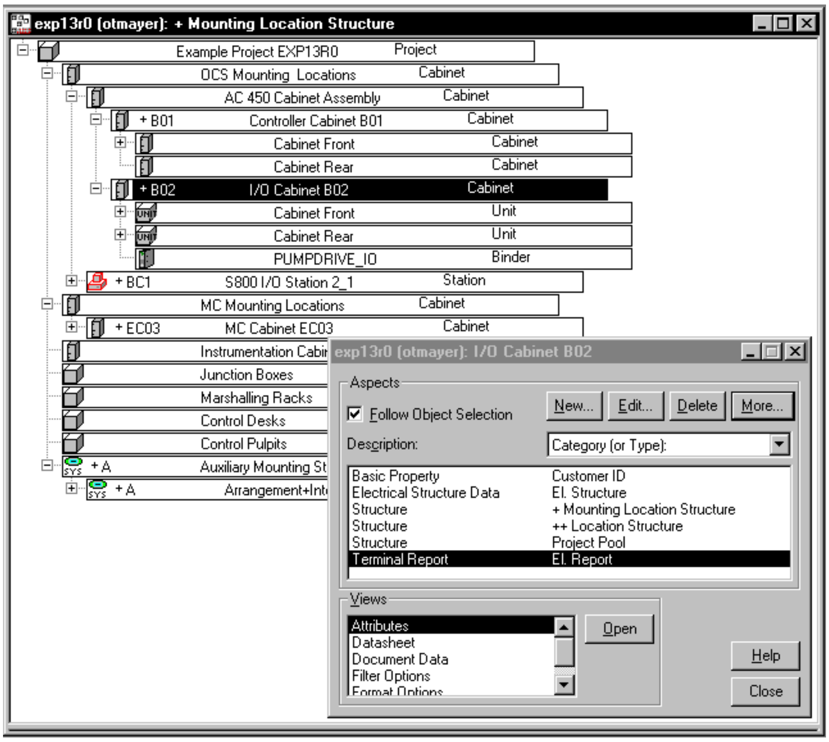
The following engineering data is assigned to objects representing mounting locations:

- Arrangement and interconnection data of the devices located in mounting locations
- Assembly data of the mounting location itself
- Data of auxiliary systems as for example the power supply items for a cabinet.

This data is mainly presented as lists, tables and diagrams. Examples are:

- Layout Diagram (Arrangement Drawing)
- Overview Diagram (for example for arrangement and interconnection)
- Circuit Diagram (for example for power supply)
- Interconnection Table
- Terminal Diagram/Table
- Assembly Drawing.
- Reports.

Figure S18-4 shows an open object window of an OCS cabinet with the assigned aspects.


exp13r0 (otmayer): + Mounting Location Structure
Example Project EXP13R0 Project
OCS Mounting Locations Cabinet
AC 450 Cabinet Assembly Cabinet
+ B01 Controller Cabinet B01 Cabinet
Cabinet Front Cabinet
Cabinet Rear Cabinet
+ B02 I/O Cabinet B02 Cabinet
Cabinet Front Unit
Cabinet Rear Unit
PUMPDRIVE_IO Binder
+ BC1 S800 I/O Station 2_1 Station
MC Mounting Locations Cabinet
+ EC03 MC Cabinet EC03 Cabinet
Instrumentation Cabir
Junction Boxes
Marshalling Racks
Control Desks
Control Pulpits
+ A Auxiliary Mounting St
+ A Arrangement+Int
exp13r0 (otmayer): I/O Cabinet B02
Aspects
New...
Edit...
Delete
More...
Views
Open
Help
Close

☑ Follow Object Selection

Description:	Category (or Type):
Basic Property	Customer ID
Electrical Structure Data	El. Structure
Structure	+ Mounting Location Structure
Structure	++ Location Structure
Structure	Project Pool
Terminal Report	El. Report

Views:
- Attributes
- Datasheet
- Document Data
- Filter Options
- Format Options

Figure S18-4. Object Window of an OCS-Cabinet

S18.1.3 Building a Mounting Location Structure from Diagrams

Structures in the Structure Builder can be built up from object information in the database of the El. Diagram Builder.

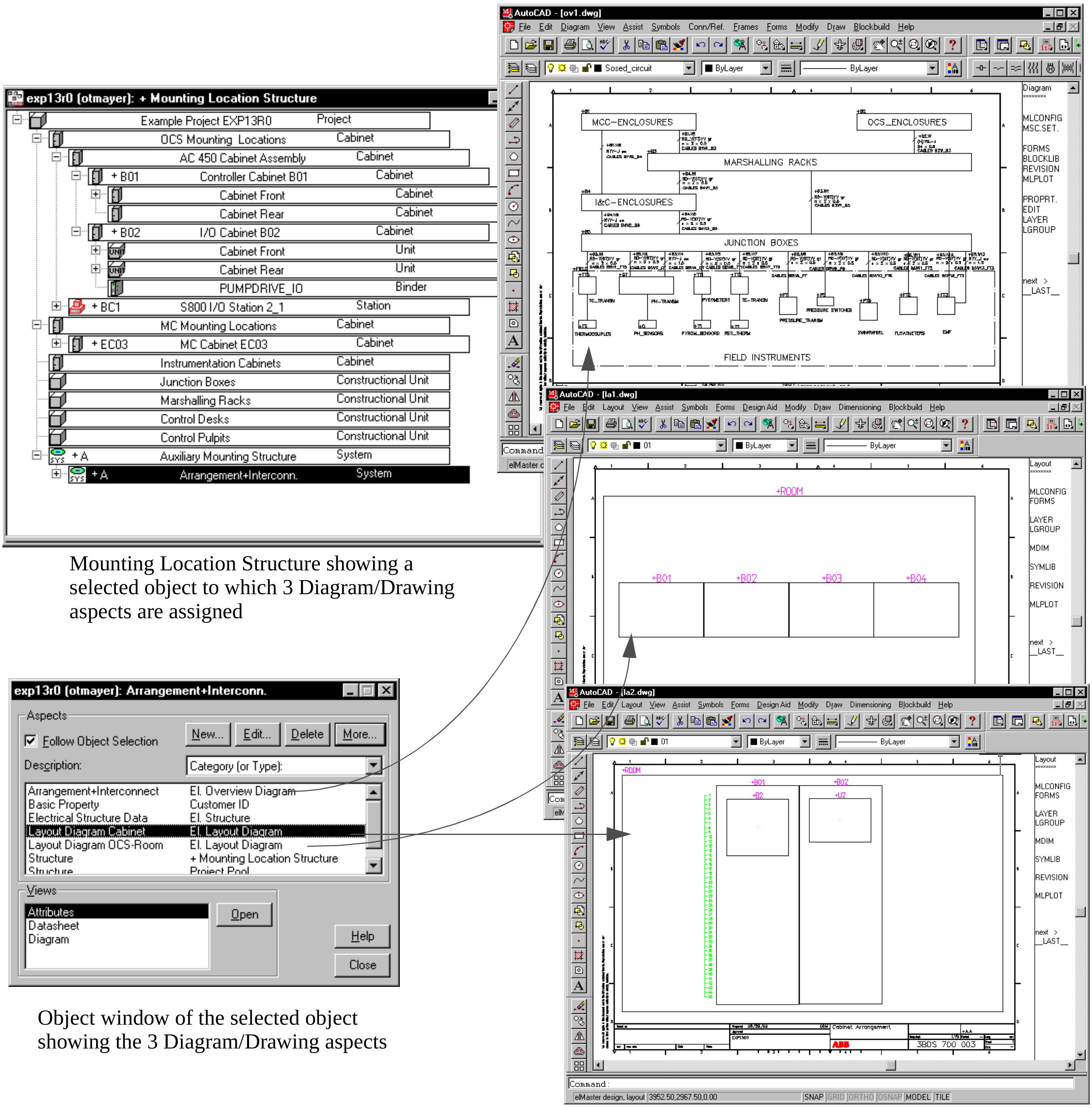
To demonstrate this we have inserted into the Mounting Location Structure of the Example Project an object to which an Overview Diagram and two Layout Diagrams are assigned as aspects. The Overview Diagram defines the principles of arrangement and interconnection and the Layout Diagrams show simple examples for the arrangement of cabinets in a room and the arrangement of racks in cabinets. The objects defined in this drawings shall be inserted into the Mounting Location Structure of the Structure Builder.

The objects defined in the drawings are inserted into the database of the Electrical Diagram Builder by giving the commands **Save to Database**. This commands can be activated from the diagram view of the Electrical Diagram Builder.

The structure information of the Structure Builder shall now be completed with structure information from the Electrical Diagram Builder database. This will be performed by the command **Synchronize STB** which can be activated from an Electrical Structure aspect in the Structure Builder.

Creating Objects by Diagrams, for Example Mounting Location Objects	
User activities:	1. Insert Diagram aspect(s) to object(s) in the Structure Builder 2. Create the new objects in the Diagram(s) 3. Assign the reference designations to the new objects 4. Save Diagram to the database of the El. Diagram Builder 5. Perform **Synchronize STB** via an El Structure aspect in the Structure Builder
Result:	Objects defined in Drawing(s) are inserted into the databases of the El. Diagram Builder and the Structure Builder

These steps are illustrated by the following figures.



exp13r0 (otmayer): + Mounting Location Structure

Example Project EXP13R0	Project
OCS Mounting Locations	Cabinet
AC 450 Cabinet Assembly	Cabinet
+ B01 Controller Cabinet B01	Cabinet
Cabinet Front	Cabinet
Cabinet Rear	Cabinet
+ B02 I/O Cabinet B02	Cabinet
Cabinet Front	Unit
Cabinet Rear	Unit
PUMPDRIVE_IO	Binder
+ BC1 S800 I/O Station 2_1	Station
MC Mounting Locations	Cabinet
+ EC03 MC Cabinet EC03	Cabinet
Instrumentation Cabinets	Cabinet
Junction Boxes	Constructional Unit
Marshalling Racks	Constructional Unit
Control Desks	Constructional Unit
Control Pulpits	Constructional Unit
+ A Auxiliary Mounting Structure	System
+ A Arrangement+Interconn.	System

Mounting Location Structure showing a selected object to which 3 Diagram/Drawing aspects are assigned

exp13r0 (otmayer): Arrangement+Interconn.

Aspects

☑ Follow Object Selection

Description: Category (or Type):

Arrangement+Interconnect	El. Overview Diagram
Basic Property	Customer ID
Electrical Structure Data	El. Structure
Layout Diagram Cabinet	El. Layout Diagram
Layout Diagram OCS-Room	El. Layout Diagram
Structure	+ Mounting Location Structure
Structure	Project Pool

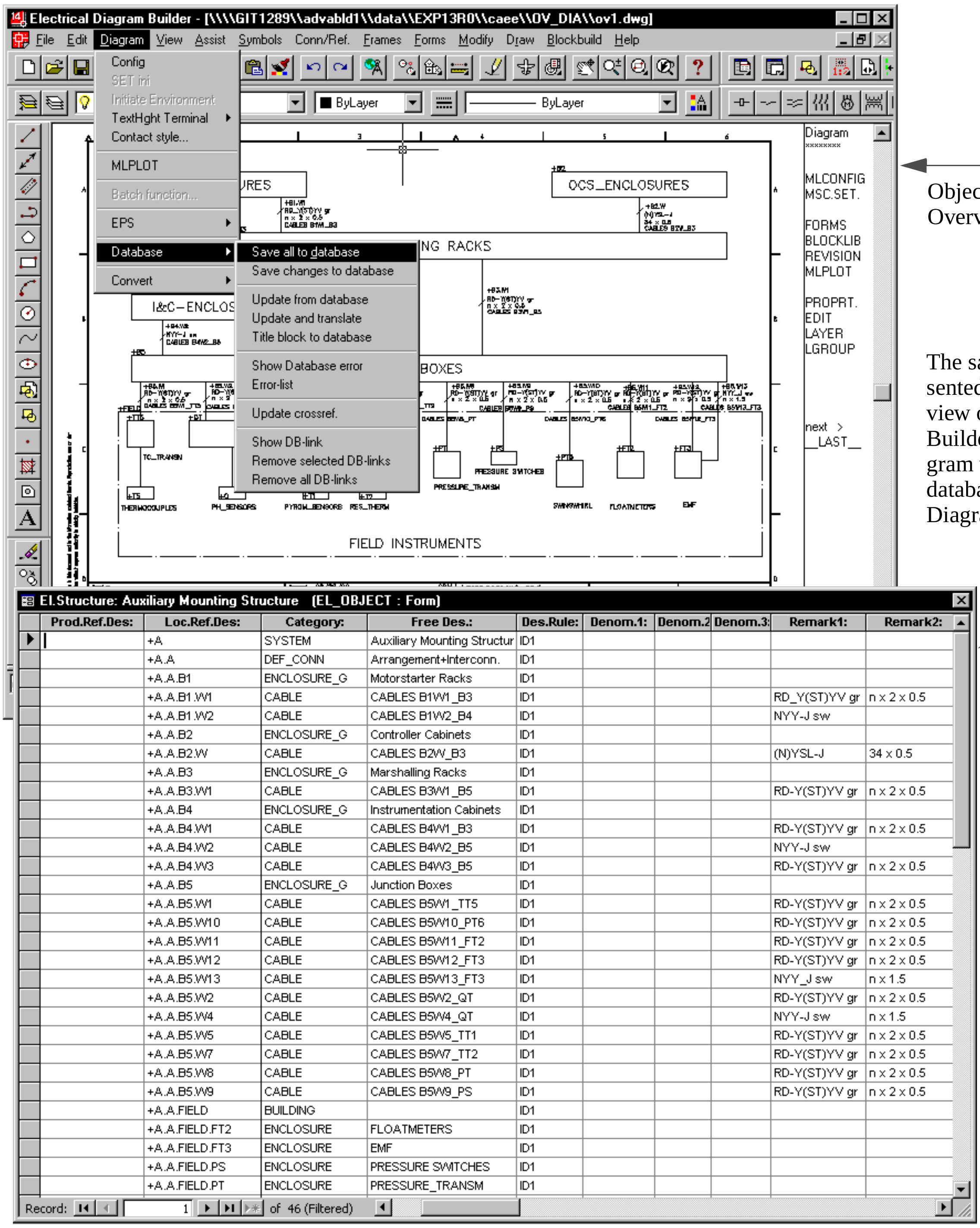
Views

Attributes
Datasheet
Diagram

Object window of the selected object showing the 3 Diagram/Drawing aspects

Diagram views of the 3 Diagram/Drawing aspects

Figure S18-5. Examples for Diagram Aspects Assigned to an Object



El.Structure: Auxiliary Mounting Structure (EL_OBJECT : Form)

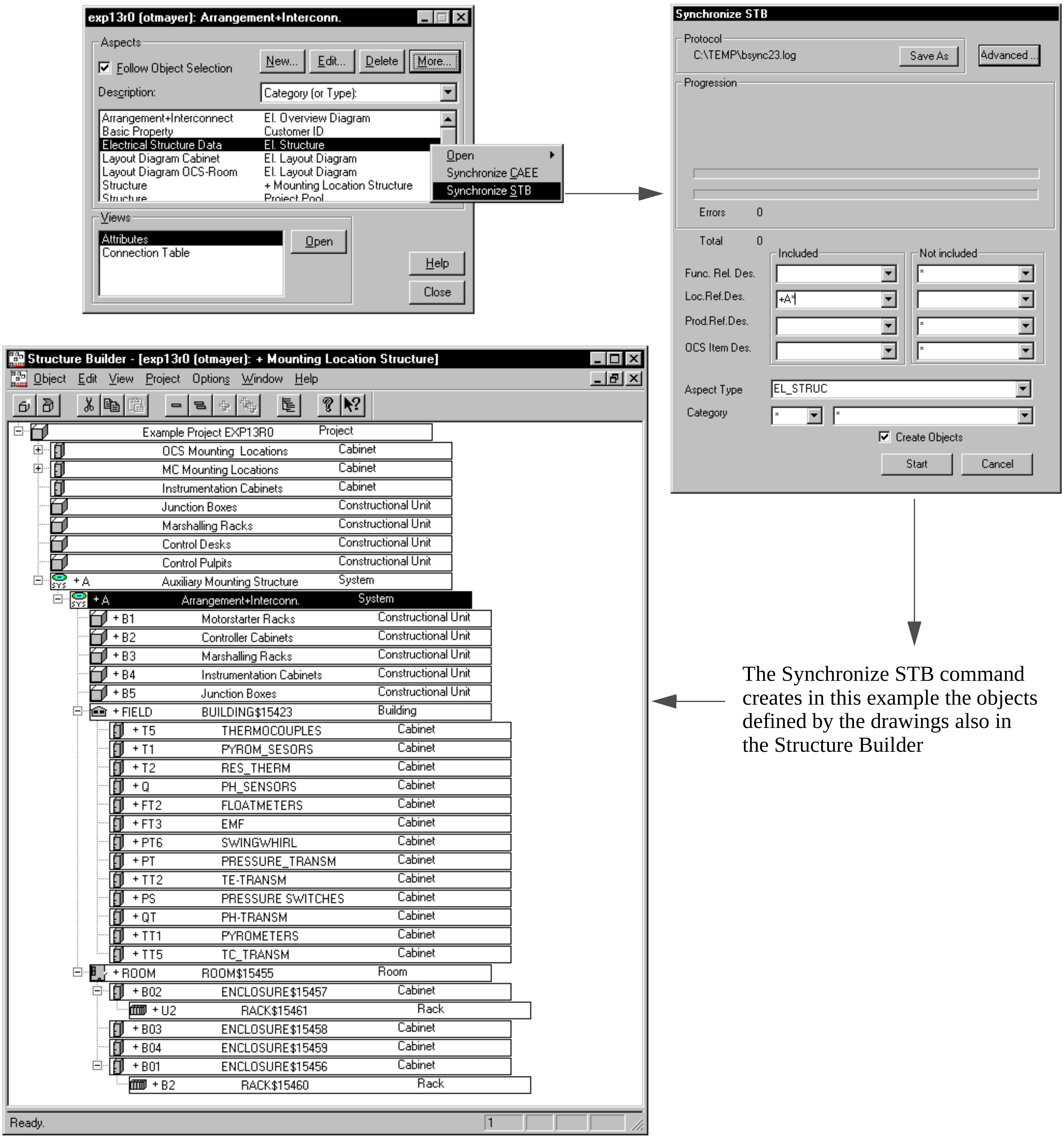
Prod.Ref.Des:	Loc.Ref.Des:	Category:	Free Des.:	Des.Rule:	Denom.1:	Denom.2	Denom.3:	Remark1:	Remark2:
	+A	SYSTEM	Auxiliary Mounting Structur	ID1					
	+A.A	DEF_CONN	Arrangement+Interconn.	ID1					
	+A.A.B1	ENCLOSURE_G	Motorstarter Racks	ID1					
	+A.A.B1.W1	CABLE	CABLES B1W1_B3	ID1				RD_Y(ST)YV gr	n x 2 x 0.5
	+A.A.B1.W2	CABLE	CABLES B1W2_B4	ID1				NYY-J sw	
	+A.A.B2	ENCLOSURE_G	Controller Cabinets	ID1					
	+A.A.B2.W	CABLE	CABLES B2W_B3	ID1				(N)YSL-J	34 x 0.5
	+A.A.B3	ENCLOSURE_G	Marshalling Racks	ID1					
	+A.A.B3.W1	CABLE	CABLES B3W1_B5	ID1				RD-Y(ST)YV gr	n x 2 x 0.5
	+A.A.B4	ENCLOSURE_G	Instrumentation Cabinets	ID1					
	+A.A.B4.W1	CABLE	CABLES B4W1_B3	ID1				RD-Y(ST)YV gr	n x 2 x 0.5
	+A.A.B4.W2	CABLE	CABLES B4W2_B5	ID1				NYY-J sw	
	+A.A.B4.W3	CABLE	CABLES B4W3_B5	ID1				RD-Y(ST)YV gr	n x 2 x 0.5
	+A.A.B5	ENCLOSURE_G	Junction Boxes	ID1					
	+A.A.B5.W1	CABLE	CABLES B5W1_TT5	ID1				RD-Y(ST)YV gr	n x 2 x 0.5
	+A.A.B5.W10	CABLE	CABLES B5W10_PT6	ID1				RD-Y(ST)YV gr	n x 2 x 0.5
	+A.A.B5.W11	CABLE	CABLES B5W11_FT2	ID1				RD-Y(ST)YV gr	n x 2 x 0.5
	+A.A.B5.W12	CABLE	CABLES B5W12_FT3	ID1				RD-Y(ST)YV gr	n x 2 x 0.5
	+A.A.B5.W13	CABLE	CABLES B5W13_FT3	ID1				NYY_J sw	n x 1.5
	+A.A.B5.W2	CABLE	CABLES B5W2_QT	ID1				RD-Y(ST)YV gr	n x 2 x 0.5
	+A.A.B5.W4	CABLE	CABLES B5W4_QT	ID1				NYY-J sw	n x 1.5
	+A.A.B5.W5	CABLE	CABLES B5W5_TT1	ID1				RD-Y(ST)YV gr	n x 2 x 0.5
	+A.A.B5.W7	CABLE	CABLES B5W7_TT2	ID1				RD-Y(ST)YV gr	n x 2 x 0.5
	+A.A.B5.W8	CABLE	CABLES B5W8_PT	ID1				RD-Y(ST)YV gr	n x 2 x 0.5
	+A.A.B5.W9	CABLE	CABLES B5W9_PS	ID1				RD-Y(ST)YV gr	n x 2 x 0.5
	+A.A.FIELD	BUILDING		ID1					
	+A.A.FIELD.FT2	ENCLOSURE	FLOATMETERS	ID1					
	+A.A.FIELD.FT3	ENCLOSURE	EMF	ID1					
	+A.A.FIELD.PS	ENCLOSURE	PRESSURE SWITCHES	ID1					
	+A.A.FIELD.PT	ENCLOSURE	PRESSURE_TRANSM	ID1					

Record: 1 of 46 (Filtered)

Objects defined in an Overview Diagram

The same objects presented by the datasheet view of the El. Diagram Builder after the Diagram was saved to the database of the El. Diagram Builder

Figure S18-6. Saving a Diagram to the Database of the El. Diagram Builder



☑ Follow Object Selection

Description: Category (or Type):

Arrangement+Interconnect	El. Overview Diagram
Basic Property	Customer ID
Electrical Structure Data	El. Structure
Layout Diagram Cabinet	El. Layout Diagram
Layout Diagram OCS-Room	El. Layout Diagram
Structure	+ Mounting Location Structure
Structure	Project Pool

Views: Attributes, Connection Table

Protocol: C:\TEMP\bsync23.log

Errors 0

Total 0

	Included	Not included
Func. Rel. Des.		*
Loc.Ref.Des.	+A*	
Prod.Ref.Des.		*
OCS Item Des.		*

Aspect Type: EL_STRUC

Category: * *

☑ Create Objects

	Example Project EXP13R0	Project
	OCS Mounting Locations	Cabinet
	MC Mounting Locations	Cabinet
	Instrumentation Cabinets	Cabinet
	Junction Boxes	Constructional Unit
	Marshalling Racks	Constructional Unit
	Control Desks	Constructional Unit
	Control Pulpits	Constructional Unit
+ A	Auxiliary Mounting Structure	System
+ A	Arrangement+Interconn.	System
+ B1	Motorstarter Racks	Constructional Unit
+ B2	Controller Cabinets	Constructional Unit
+ B3	Marshalling Racks	Constructional Unit
+ B4	Instrumentation Cabinets	Constructional Unit
+ B5	Junction Boxes	Constructional Unit
+ FIELD	BUILDING$15423	Building
+ T5	THERMOCOUPLES	Cabinet
+ T1	PYROM_SESORS	Cabinet
+ T2	RES_THERM	Cabinet
+ Q	PH_SENSORS	Cabinet
+ FT2	FLOATMETERS	Cabinet
+ FT3	EMF	Cabinet
+ PT6	SWINGWHIRL	Cabinet
+ PT	PRESSURE_TRANSM	Cabinet
+ TT2	TE-TRANSM	Cabinet
+ PS	PRESSURE SWITCHES	Cabinet
+ QT	PH-TRANSM	Cabinet
+ TT1	PYROMETERS	Cabinet
+ TT5	TC_TRANSM	Cabinet
+ ROOM	ROOM$15455	Room
+ B02	ENCLOSURE$15457	Cabinet
+ U2	RACK$15461	Rack
+ B03	ENCLOSURE$15458	Cabinet
+ B04	ENCLOSURE$15459	Cabinet
+ B01	ENCLOSURE$15456	Cabinet
+ B2	RACK$15460	Rack

The Synchronize STB command creates in this example the objects defined by the drawings also in the Structure Builder

Figure S18-7. Synchronizing El. Diagram Builder to Structure Builder Database

Step 19 Arrange Components and Devices

S19.1 Placing of Devices and Components in Locations

Devices and Components can be placed on

- Topographic locations
- Mounting locations.

This can be performed in the Advant object model by placing Advant objects, for example, by drag and drop, to other Advant objects representing locations (topographic locations or mounting locations) in the relevant structures.

S19.2 Placing Objects of the Drive System Example

Placing the OCS Boards in the Mounting Location Structure

Figure S19-1 shows the placing of the S800 OCS-boards of the example drive system into a S800 station in the Mounting Location Structure.

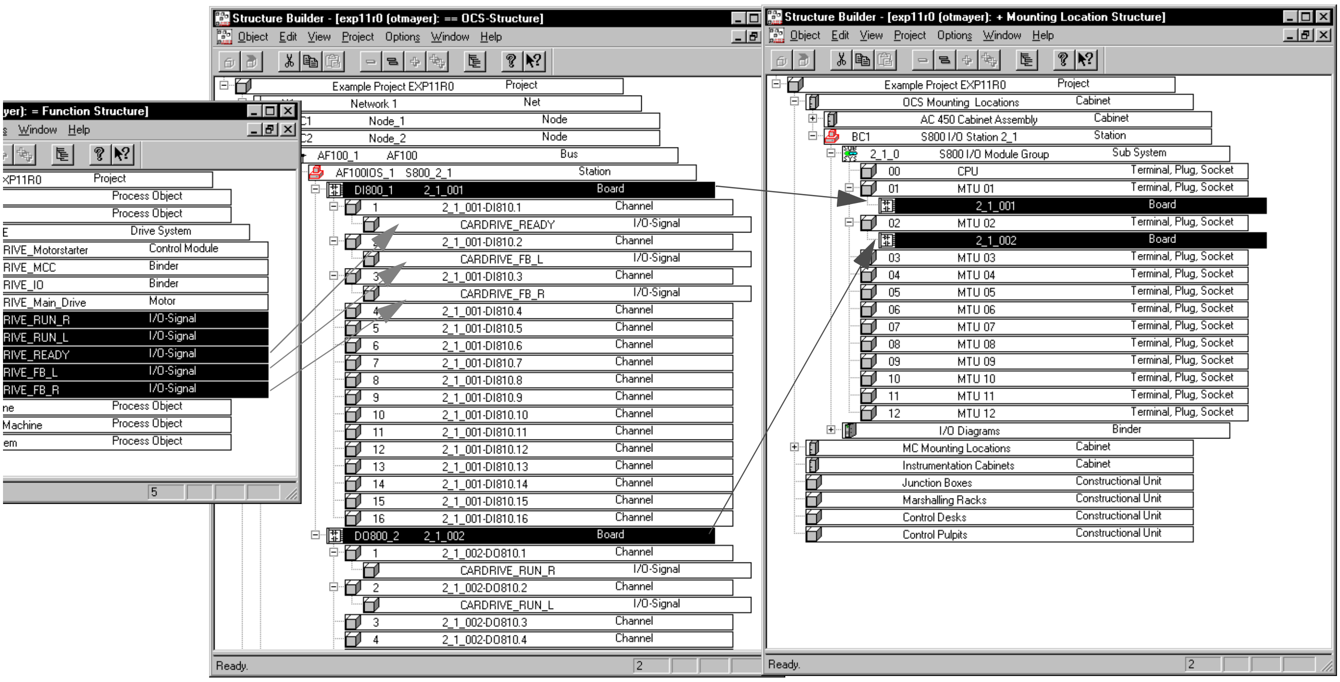
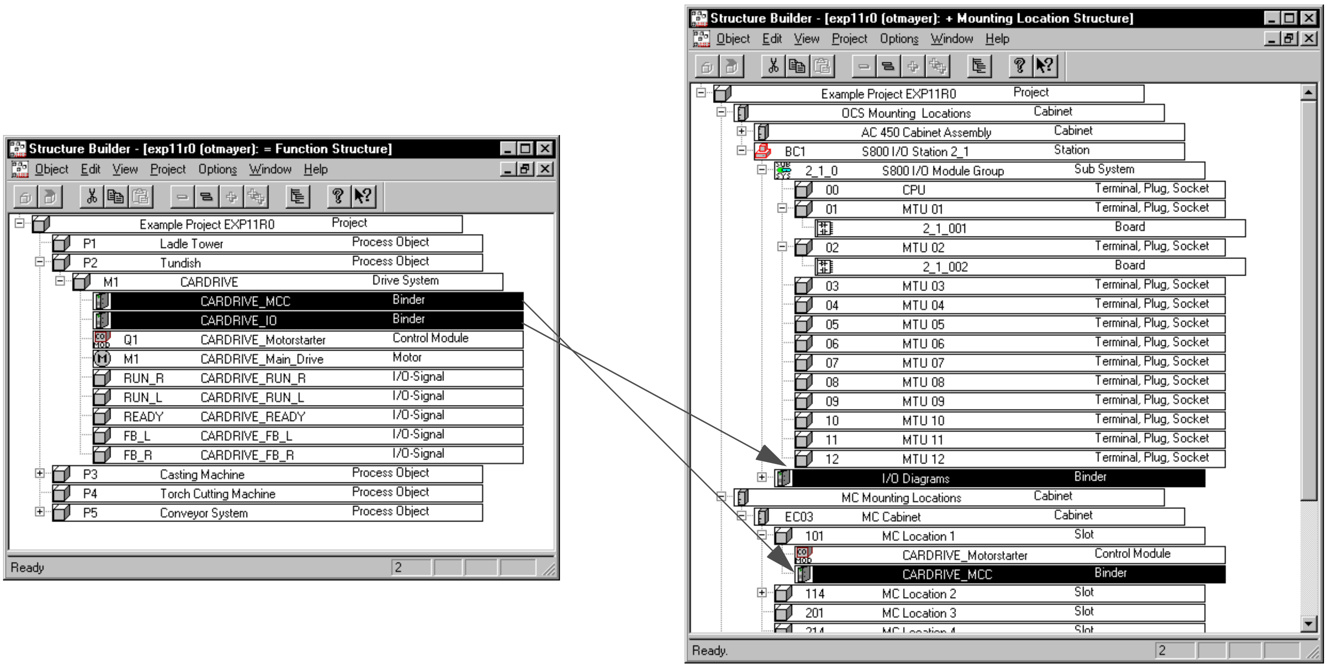

yer]: = Function Structure]
Window Help
Structure Builder - [exp11r0 (otmayer): == OCS-Structure]
Object Edit View Project Options Window Help
Structure Builder - [exp11r0 (otmayer): + Mounting Location Structure]
Object Edit View Project Options Window Help
Example Project EXP11R0 Project
Network 1 Net
Node_1 Node
Node_2 Node
AF100_1 AF100 Bus
AF100IOS_1 S800_2_1 Station
DI800_1 2_1_001 Board
2_1_001-DI810.1 Channel
CARDRIVE_READY I/O-Signal
CARDRIVE_FB_L I/O-Signal
CARDRIVE_FB_R I/O-Signal
DO800_2 2_1_002 Board
CARDRIVE_RUN_R I/O-Signal
CARDRIVE_RUN_L I/O-Signal
OCS Mounting Locations Cabinet
AC 450 Cabinet Assembly Cabinet
BC1 S800 I/O Station 2_1 Station
2_1_0 S800 I/O Module Group Sub System
00 CPU Terminal, Plug, Socket
01 MTU 01 Terminal, Plug, Socket
2_1_001 Board
02 MTU 02 Terminal, Plug, Socket
2_1_002 Board
I/O Diagrams Binder
MC Mounting Locations Cabinet
Instrumentation Cabinets Cabinet
Junction Boxes Constructional Unit
Marshalling Racks Constructional Unit
Control Desks Constructional Unit
Control Pulpits Constructional Unit
Ready.

Figure S19-1. Placing the OCS-Boards in the Mounting Location Structure

Placing the Binders in the Mounting Location Structure

Figure S19-2 shows the placing of the Binders with the Circuit Diagrams into the Mounting Location Structure. Place the binders in such a way that the binders get an absolute reference designation that fits as designation of the diagrams shown in the title blocks. In our example the I/O Circuit Diagrams are assigned to the S800 I/O station to which also the relevant boards for the signals of the example are assigned. The binder with the Circuit Diagrams of the motor and motorstarter is assigned to the motor controller cabinet in which the motorstarter is placed.


Structure Builder - [exp11r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
Ready
2
Structure Builder - [exp11r0 (otmayer): + Mounting Location Structure]
Object Edit View Project Options Window Help
Ready.
2

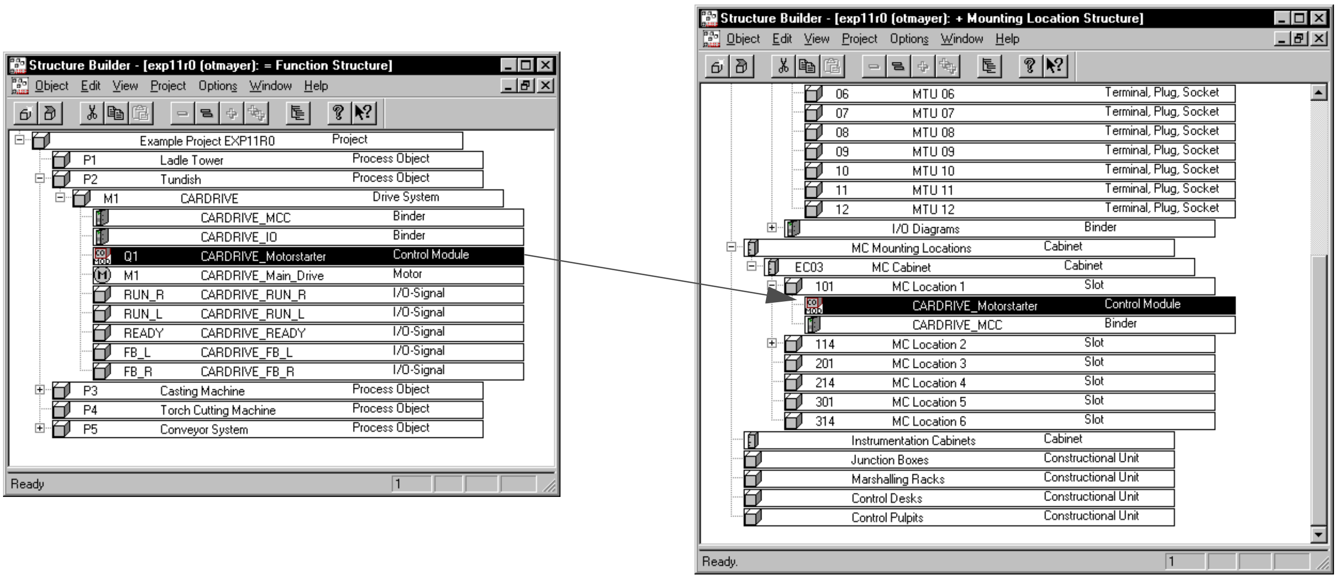
	Example Project EXP11R0	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
M1	CARDRIVE	Drive System
	CARDRIVE_MCC	Binder
	CARDRIVE_IO	Binder
Q1	CARDRIVE_Motorstarter	Control Module
M1	CARDRIVE_Main_Drive	Motor
RUN_R	CARDRIVE_RUN_R	I/O-Signal
RUN_L	CARDRIVE_RUN_L	I/O-Signal
READY	CARDRIVE_READY	I/O-Signal
FB_L	CARDRIVE_FB_L	I/O-Signal
FB_R	CARDRIVE_FB_R	I/O-Signal
P3	Casting Machine	Process Object
P4	Torch Cutting Machine	Process Object
P5	Conveyor System	Process Object

	Example Project EXP11R0	Project
	OCS Mounting Locations	Cabinet
	AC 450 Cabinet Assembly	Cabinet
BC1	S800 I/O Station 2_1	Station
2_1_0	S800 I/O Module Group	Sub System
00	CPU	Terminal, Plug, Socket
01	MTU 01	Terminal, Plug, Socket
	2_1_001	Board
02	MTU 02	Terminal, Plug, Socket
	2_1_002	Board
03	MTU 03	Terminal, Plug, Socket
04	MTU 04	Terminal, Plug, Socket
05	MTU 05	Terminal, Plug, Socket
06	MTU 06	Terminal, Plug, Socket
07	MTU 07	Terminal, Plug, Socket
08	MTU 08	Terminal, Plug, Socket
09	MTU 09	Terminal, Plug, Socket
10	MTU 10	Terminal, Plug, Socket
11	MTU 11	Terminal, Plug, Socket
12	MTU 12	Terminal, Plug, Socket
	I/O Diagrams	Binder
	MC Mounting Locations	Cabinet
EC03	MC Cabinet	Cabinet
101	MC Location 1	Slot
	CARDRIVE_Motorstarter	Control Module
	CARDRIVE_MCC	Binder
114	MC Location 2	Slot
201	MC Location 3	Slot
214	MC Location 4	Slot

Figure S19-2. Placing of Binders in the Mounting Location Structure

Placing the Motorstarter in a MC Cabinet

Figure S19-3 shows the placing of the motorstarter into a slot of a motor controller cabinet.


Structure Builder - [exp11r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
Ready
1
Structure Builder - [exp11r0 (otmayer): + Mounting Location Structure]
Object Edit View Project Options Window Help
Ready.
1

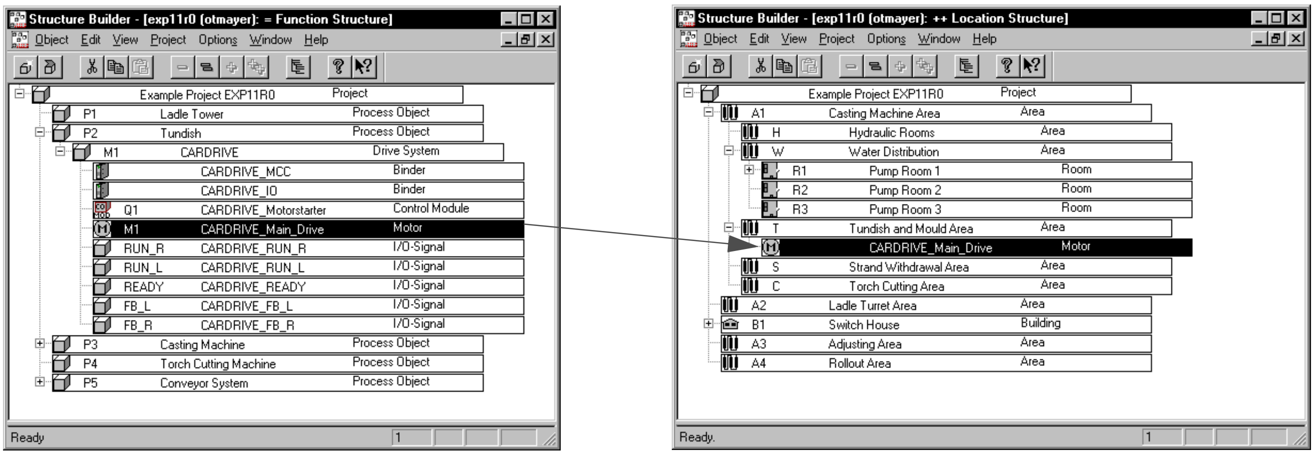
	Example Project EXP11R0	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
M1	CARDRIVE	Drive System
	CARDRIVE_MCC	Binder
	CARDRIVE_IO	Binder
Q1	CARDRIVE_Motorstarter	Control Module
M1	CARDRIVE_Main_Drive	Motor
RUN_R	CARDRIVE_RUN_R	I/O-Signal
RUN_L	CARDRIVE_RUN_L	I/O-Signal
READY	CARDRIVE_READY	I/O-Signal
FB_L	CARDRIVE_FB_L	I/O-Signal
FB_R	CARDRIVE_FB_R	I/O-Signal
P3	Casting Machine	Process Object
P4	Torch Cutting Machine	Process Object
P5	Conveyor System	Process Object

06	MTU 06	Terminal, Plug, Socket
07	MTU 07	Terminal, Plug, Socket
08	MTU 08	Terminal, Plug, Socket
09	MTU 09	Terminal, Plug, Socket
10	MTU 10	Terminal, Plug, Socket
11	MTU 11	Terminal, Plug, Socket
12	MTU 12	Terminal, Plug, Socket
	I/O Diagrams	Binder
	MC Mounting Locations	Cabinet
EC03	MC Cabinet	Cabinet
101	MC Location 1	Slot
	CARDRIVE_Motorstarter	Control Module
	CARDRIVE_MCC	Binder
114	MC Location 2	Slot
201	MC Location 3	Slot
214	MC Location 4	Slot
301	MC Location 5	Slot
314	MC Location 6	Slot
	Instrumentation Cabinets	Cabinet
	Junction Boxes	Constructional Unit
	Marshalling Racks	Constructional Unit
	Control Desks	Constructional Unit
	Control Pulpits	Constructional Unit

Figure S19-3. Placing the Motorcontroller in a Cabinet

Placing the Motor in the Topographic Location Structure

Figure S19-4 shows the placing of the motor in the Topographic Location structure.


Structure Builder - [exp11r0 (otmayer): = Function Structure]
Object Edit View Project Options Window Help
Ready
1
Structure Builder - [exp11r0 (otmayer): ++ Location Structure]
Object Edit View Project Options Window Help
Ready.
1

	Example Project EXP11R0	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
M1	CARDRIVE	Drive System
	CARDRIVE_MCC	Binder
	CARDRIVE_IO	Binder
Q1	CARDRIVE_Motorstarter	Control Module
M1	CARDRIVE_Main_Drive	Motor
RUN_R	CARDRIVE_RUN_R	I/O-Signal
RUN_L	CARDRIVE_RUN_L	I/O-Signal
READY	CARDRIVE_READY	I/O-Signal
FB_L	CARDRIVE_FB_L	I/O-Signal
FB_R	CARDRIVE_FB_R	I/O-Signal
P3	Casting Machine	Process Object
P4	Torch Cutting Machine	Process Object
P5	Conveyor System	Process Object

	Example Project EXP11R0	Project
A1	Casting Machine Area	Area
H	Hydraulic Rooms	Area
W	Water Distribution	Area
R1	Pump Room 1	Room
R2	Pump Room 2	Room
R3	Pump Room 3	Room
T	Tundish and Mould Area	Area
	CARDRIVE_Main_Drive	Motor
S	Strand Withdrawal Area	Area
C	Torch Cutting Area	Area
A2	Ladle Turret Area	Area
B1	Switch House	Building
A3	Adjusting Area	Area
A4	Rollout Area	Area

Figure S19-4. Placing the Motor in the Topographic Location Structure

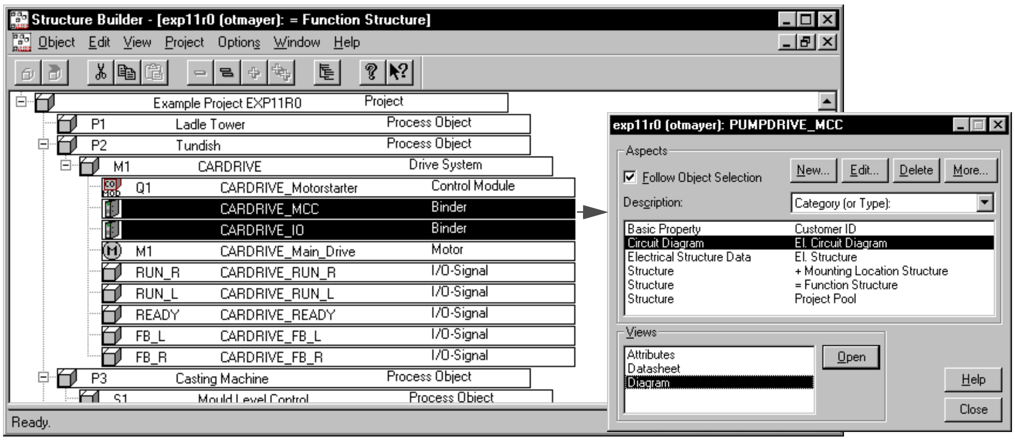
Step 20 Build/Adapt Circuit Diagrams

S20.1 Create Electrical Circuit Diagram

Circuit diagrams are assigned to El. Circuit Diagram aspects of Advant objects. It is recommended to use a separate Advant object of type *Binder* to get the full flexibility in positioning circuit diagrams in the structures**.** A circuit diagram gets its designation dependent on the position of its carrier object (the Binder object) in the Function Structure and in the Mounting Location Structure. The absolute reference designation of the carrier object in the Function Structure and in the Mounting Location Structure is inserted automatically into the corresponding footer fields of the circuit diagram.

Figure S20-1 shows the location of the binder objects of a drive system in the Function Structure and an open aspect window of one Binder object.

Creation of a Circuit Diagram Aspect	
Preconditions:	Object of type **Binder** created
User activities:	1. Use the Structure Builder 2. Open the Object window of the Binder object to which an El. Circuit Diagram shall be assigned and click on **New** 3. Select **El. Circuit Diagram** in the **New Aspect** window and fill in the **Aspect Description** 4. Click on **OK**
Result:	El. Circuit Diagram aspect created



	Example Project EXP11R0	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
M1	CARDRIVE	Drive System
Q1	CARDRIVE_Motorstarter	Control Module
	CARDRIVE_MCC	Binder
	CARDRIVE_IO	Binder
M1	CARDRIVE_Main_Drive	Motor
RUN_R	CARDRIVE_RUN_R	I/O-Signal
RUN_L	CARDRIVE_RUN_L	I/O-Signal
READY	CARDRIVE_READY	I/O-Signal
FB_L	CARDRIVE_FB_L	I/O-Signal
FB_R	CARDRIVE_FB_R	I/O-Signal
P3	Casting Machine	Process Object
S1	Mould Level Control	Process Object

Basic Property	Customer ID
Circuit Diagram	El. Circuit Diagram
Electrical Structure Data	El. Structure
Structure	+ Mounting Location Structure
Structure	= Function Structure
Structure	Project Pool

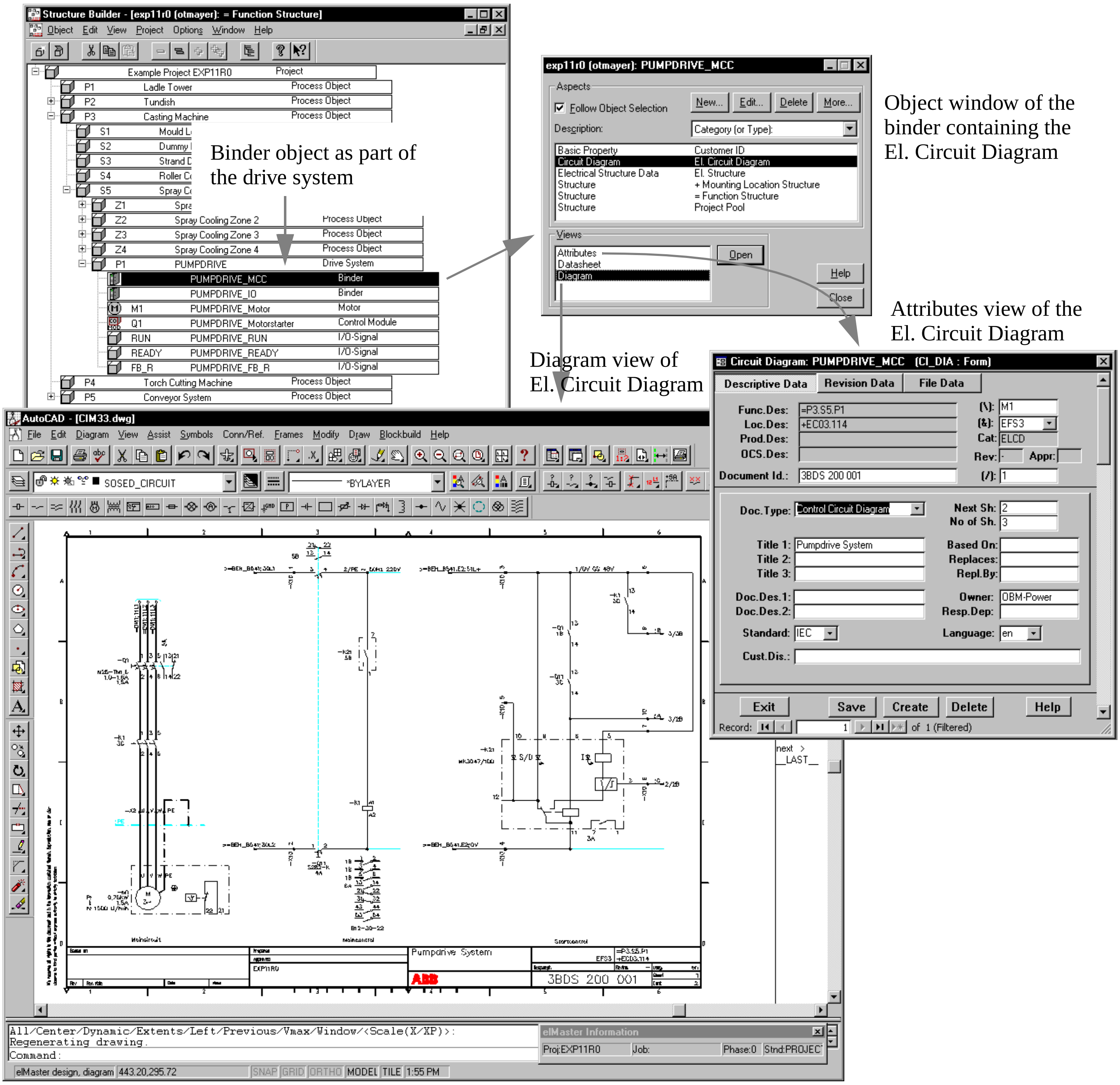
Attributes
Datasheet
Diagram

Figure S20-1. Binder Objects in the Function Structure

S20.2 Edit Electrical Circuit Diagram

Editing of new created or of imported typical circuit diagrams is performed by opening the diagram view on the corresponding aspect and editing it with the Electrical Diagram Builder.

Figure S20-2 shows the windows involved when editing an Electrical Circuit Diagram.



Binder object as part of the drive system

Object window of the binder containing the El. Circuit Diagram

Diagram view of El. Circuit Diagram

Attributes view of the El. Circuit Diagram

Structure Builder - [exp11r0 (otmayer): = Function Structure]

	Example Project EXP11R0	Project
P1	Ladle Tower	Process Object
P2	Tundish	Process Object
P3	Casting Machine	Process Object
S1	Mould L	
S2	Dummy	
S3	Strand D	
S4	Roller C	
S5	Spray C	
Z1	Spra	
Z2	Spray Cooling Zone 2	Process Object
Z3	Spray Cooling Zone 3	Process Object
Z4	Spray Cooling Zone 4	Process Object
P1	PUMPDRIVE	Drive System
	PUMPDRIVE_MCC	Binder
	PUMPDRIVE_IO	Binder
M1	PUMPDRIVE_Motor	Motor
Q1	PUMPDRIVE_Motorstarter	Control Module
RUN	PUMPDRIVE_RUN	I/O-Signal
READY	PUMPDRIVE_READY	I/O-Signal
FB_R	PUMPDRIVE_FB_R	I/O-Signal
P4	Torch Cutting Machine	Process Object
P5	Conveyor System	Process Object

exp11r0 (otmayer): PUMPDRIVE_MCC

Aspects

☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
Circuit Diagram	El. Circuit Diagram
Electrical Structure Data	El. Structure
Structure	+ Mounting Location Structure
Structure	= Function Structure
Structure	Project Pool

Views

- Attributes
- Datasheet
- Diagram

AutoCAD - [CIM33.dwg]

```
All/Center/Dynamic/Extents/Left/Previous/Vmax/Window/<Scale(X/XP)>:
Regenerating drawing.
Command:
```

elMaster design, diagram 443.20,295.72 SNAP GRID ORTHO MODEL TILE 1:55 PM

elMaster Information

Proj:EXP11R0 Job: Phase:0 Stnd:PROJEC

Circuit Diagram: PUMPDRIVE_MCC (CI_DIA : Form)

Descriptive Data | Revision Data | File Data

Func.Des: =P3.S5.P1
Loc.Des: +EC03.114
Prod.Des:
OCS.Des:
Document Id.: 3BDS 200 001
(\): M1
(&): EFS3
Cat: ELCD
Rev: -
Appr:
(/): 1
Doc.Type: Control Circuit Diagram
Next Sh: 2
No of Sh.: 3
Title 1: Pumpdrive System
Title 2:
Title 3:
Based On:
Replaces:
Repl.By:
Doc.Des.1:
Doc.Des.2:
Owner: OBM-Power
Resp.Dep:
Standard: IEC
Language: en
Cust.Dis.:

Record: 1 of 1 (Filtered)

Object Edit View Project Options Window Help
New... Edit... Delete More...
Open
Help
Close
File Edit Diagram View Assist Symbols Conn/Ref. Frames Modify Draw Blockbuild Help
SOSED_CIRCUIT
*BYLAYER
Pumpdrive System
EFS3
=P3.S5.P1
+EC03.114
EXP11R0
3BDS 200 001
ABB
next >
__LAST__
Exit
Save
Create
Delete
Help

Figure S20-2. Editing an Electrical Circuit Diagram

Step 21 Update Circuit Diagrams

S21.1 Preconditions

When you have completed the object aspect related engineering of a project or part of a project, you can initiate the steps which perform the final update of the circuit diagrams. The preconditions for these steps are

- Objects and object aspects are completely defined
- Objects are placed in the proper structure positions
- Designations and relative reference designations are assigned.

The update procedure completes the following information in Circuit Diagrams

- Designations
- Reference Designations
- Cross References.

The following table gives an overview about the steps which perform the final update of circuit diagrams. These steps are described in the following sections.

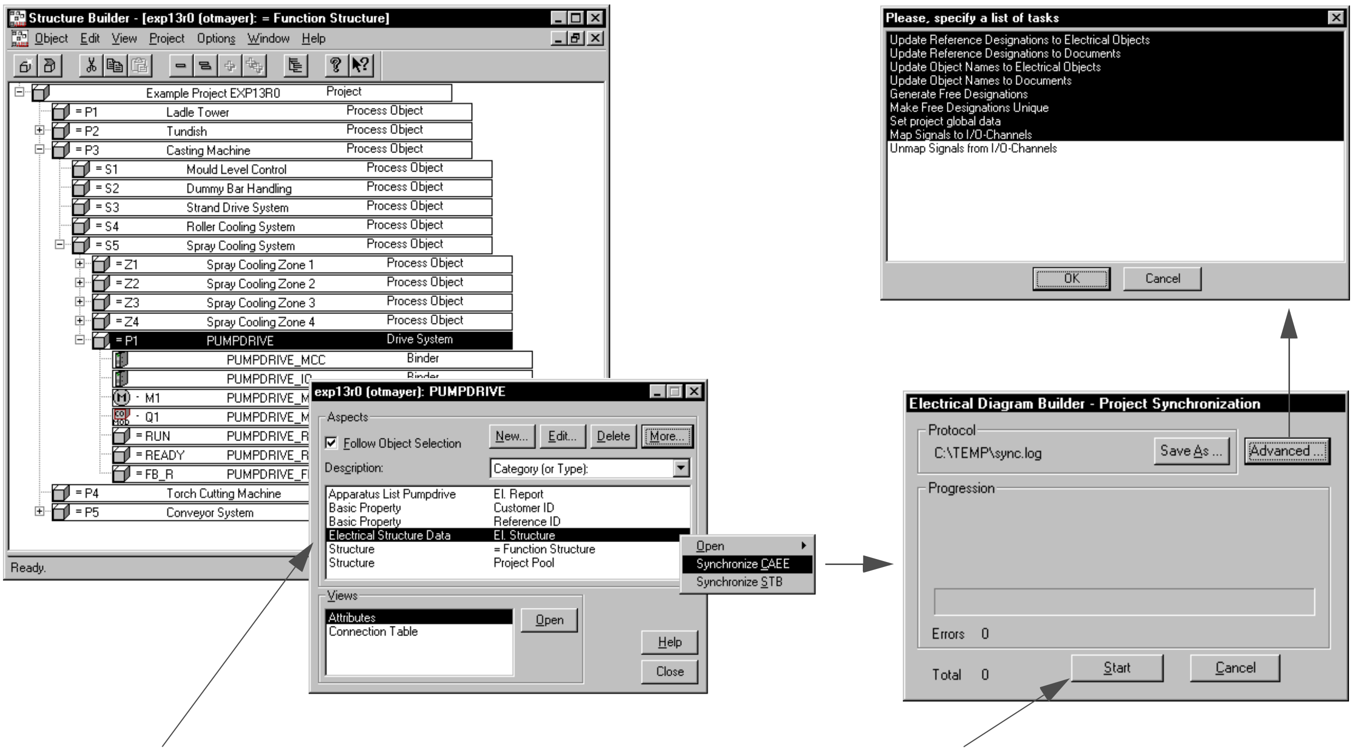
Update of Circuit Diagrams	
User activities:	1. Synchronize Structure Builder - Electrical Diagram Builder by the command Synchronize CAEE 2. Save Board/Connection Unit Circuit Diagrams to database 3. Save I/O Circuit Diagrams to database 4. Save Consumer Circuit Diagrams to database 5. Synchronize again Structure Builder - Electrical Diagram Builder by the command **Synchronize CAEE** 6. Update Board/Connection Unit Circuit Diagrams from database 7. Update I/O Circuit Diagrams from database 8. Update Consumer Circuit Diagrams from database
Result:	Circuit Diagrams are updated with regard to designations, cross-references and technical information.

S21.2 Synchronization Structure Builder-Electrical Diagram Builder

Before you save and update circuit diagrams you must synchronize Structure Builder and Electrical Diagram Builder to transfer all structure and designation information defined by the

Structure Builder into the Electrical Diagram Builder database. Before you give the final synchronization command you must have placed the relevant objects into their final positions in the Function Structure, the OCS-Structure and the Mounting Location Structure. For example the Binder objects keeping the electrical circuit diagrams of our pumpdrive example were placed in the Function Structure and in the Mounting Location Structure. The circuit diagrams in these Binder objects get their designations in the title block from the absolute reference designation of the Binder objects according their position.

The synchronization can be performed on any Electrical Structure aspect of an object in the Structure Builder. Figure S21-1 shows the initialization of the synchronization of a complete project.



Structure Builder - [exp13r0 (otmayer): = Function Structure]

Example Project EXP13R0 Project
= P1 Ladle Tower Process Object
= P2 Tundish Process Object
= P3 Casting Machine Process Object
= S1 Mould Level Control Process Object
= S2 Dummy Bar Handling Process Object
= S3 Strand Drive System Process Object
= S4 Roller Cooling System Process Object
= S5 Spray Cooling System Process Object
= Z1 Spray Cooling Zone 1 Process Object
= Z2 Spray Cooling Zone 2 Process Object
= Z3 Spray Cooling Zone 3 Process Object
= Z4 Spray Cooling Zone 4 Process Object
= P1 PUMPDRIVE Drive System
PUMPDRIVE_MCC Binder
PUMPDRIVE_IO Binder
- M1 PUMPDRIVE_M
- Q1 PUMPDRIVE_M
= RUN PUMPDRIVE_R
= READY PUMPDRIVE_R
= FB_R PUMPDRIVE_F
= P4 Torch Cutting Machine
= P5 Conveyor System

Ready.

exp13r0 (otmayer): PUMPDRIVE

Aspects

☑ Follow Object Selection

Description: Category (or Type):

Apparatus List Pumpdrive	El. Report
Basic Property	Customer ID
Basic Property	Reference ID
Electrical Structure Data	El. Structure
Structure	= Function Structure
Structure	Project Pool

Views

Attributes
Connection Table

Electrical Diagram Builder - Project Synchronization

Protocol: C:\TEMP\sync.log

Progression

Errors 0

Total 0

Please, specify a list of tasks

Update Reference Designations to Electrical Objects
Update Reference Designations to Documents
Update Object Names to Electrical Objects
Update Object Names to Documents
Generate Free Designations
Make Free Designations Unique
Set project global data
Map Signals to I/O-Channels
Unmap Signals from I/O-Channels

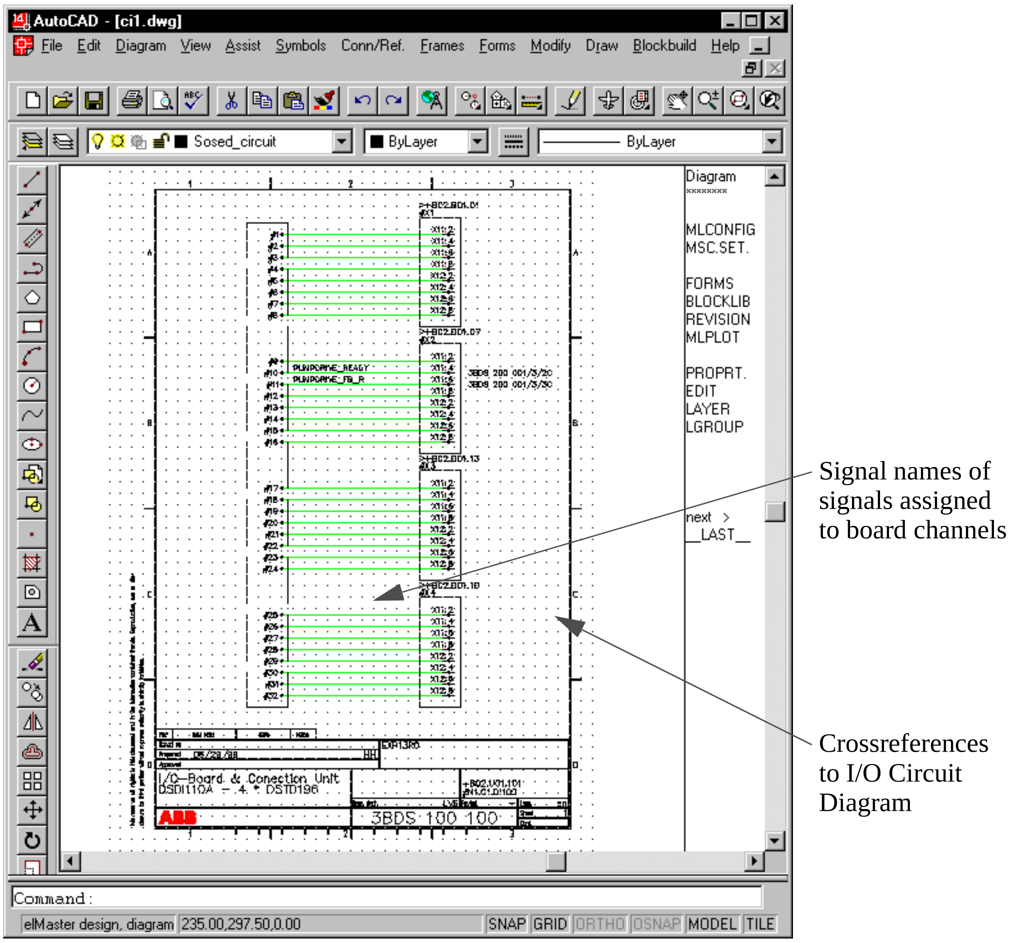
Synchronization is initiated via an El. Structure aspect of an object in the Structure Builder

The final START command for synchronization is given after selection is done in the list of tasks

Figure S21-1. Synchronizing Structure Builder to El. Diagram Builder Database

S21.2.1 Save Board/Connection Unit Circuit Diagrams to Database

When the circuit diagrams are completed, you save it to the Electrical Diagram Builder database by the **Save to Database** command. In our example of Figure S21-2 the correlations of the objects representing board channels, connection unit terminals and signals are established. The crossreferences to I/O diagrams are also shown in this diagram. They are however indicated not before saving of the relevant I/O diagrams and giving an additional update command on the drawing.



Signal names of signals assigned to board channels

Crossreferences to I/O Circuit Diagram

Figure S21-2. Circuit Diagram Board-Connection Unit

NOTE

The relationship between channels and connection unit can only be handled if the Wiring Builder option is installed.

S21.2.2 Save I/O Circuit Diagrams to Database

When the I/O circuit diagrams are completed, you save it to the Electrical Diagram Builder database by the **Save to Database** command. The object information of the objects in the diagrams are transferred by this command to the database of the Electrical Diagram Builder and will be correlated with the already existing board channel information in the database via the signal name. The board channel and the connection unit information will be assigned to the symbols in the drawings.

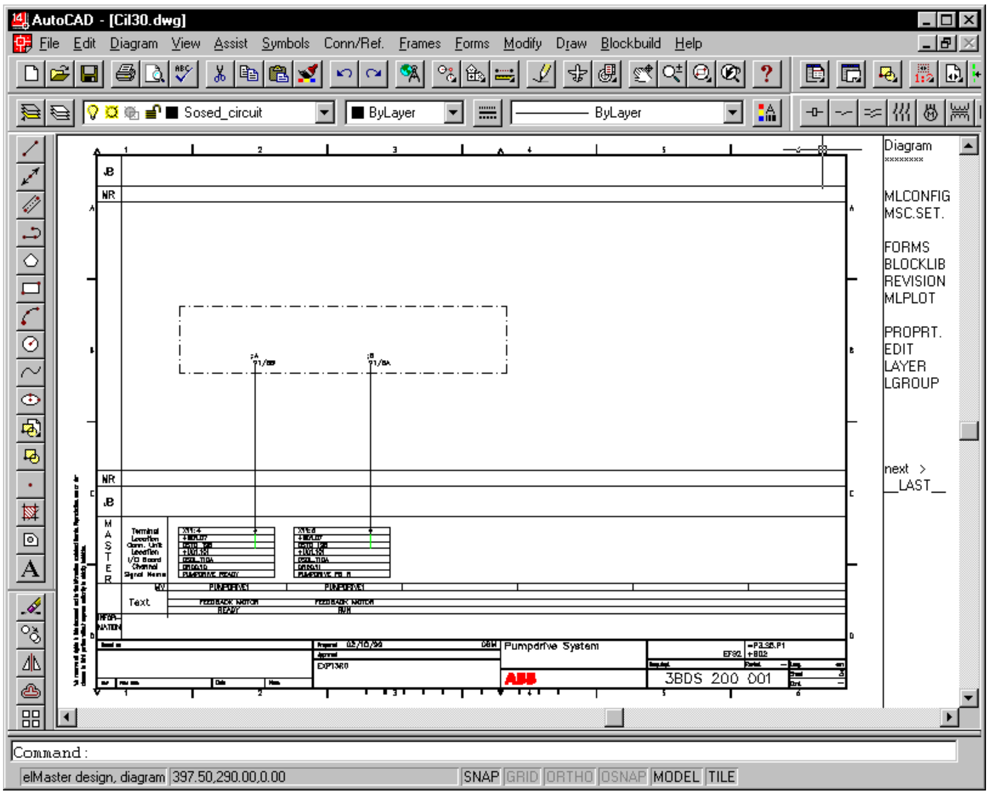


Figure S21-3. I/O Circuit Diagram Showing Two Digital Input Channels

S21.2.3 Save Circuit Diagrams for Consumers to Database

When the circuit diagrams for consumers (for example for a drive system) are completed, you save it to the Electrical Diagram Builder database by the **Save to Database** command. The object information of the objects in the drawings are transferred by this command to the database and care correlated with the already existing information in the database.

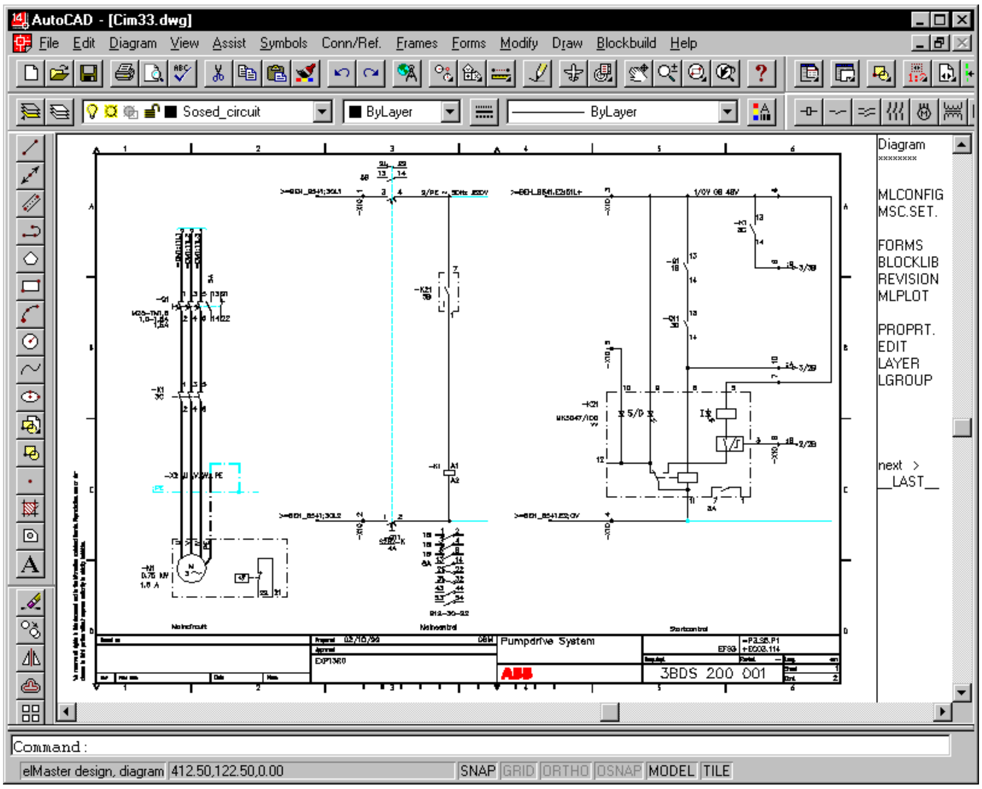


Figure S21-4. Circuit Diagram of a Drive System

S21.2.4 Synchronize CAEE

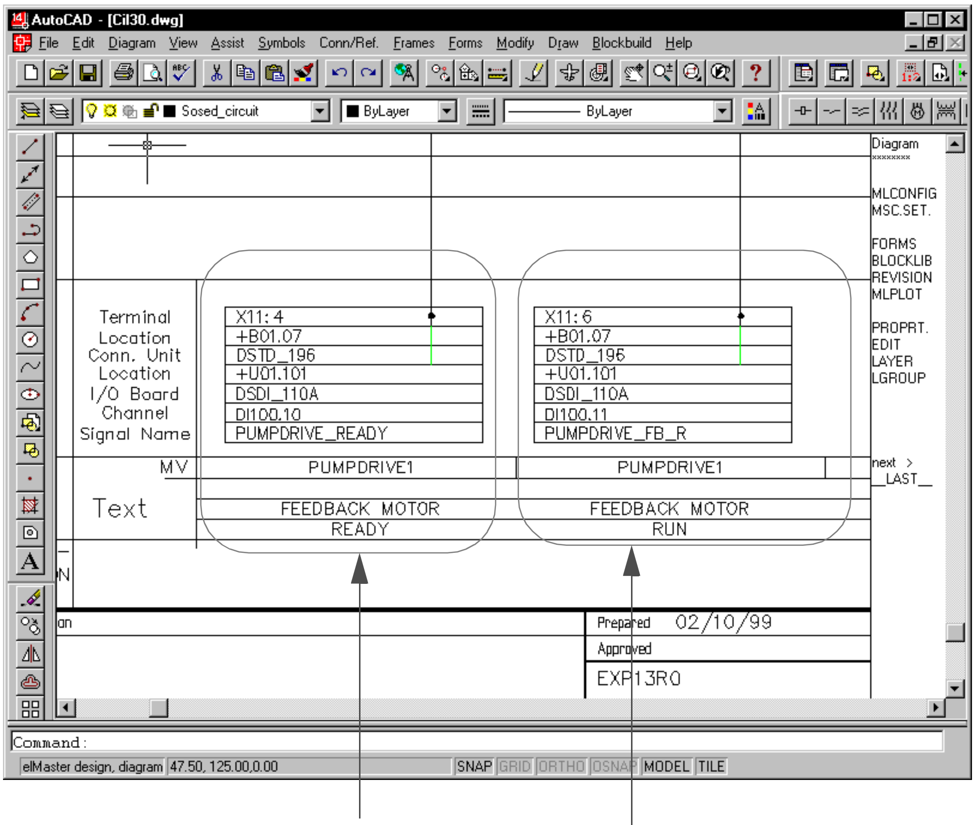
After you saved all relevant circuit diagrams to database you should again perform the command **Synchronize CAEE**. This is necessary because actual information from the Structure Builder could have been changed or overwritten.

S21.2.5 Update Circuit Diagrams Board-Connection Unit from Database

The cross references to the I/O circuit diagrams are inserted and the Mounting Location reference designation of the connection units are completed by the **Update from Database** command.

S21.2.6 Update I/O Circuit Diagrams from Database

The information related to I/O channels is completed according to the actual information in the database of the Electrical Diagram Builder by giving the **Update from Database** command.



Completion of the channel related information after save to and update from database

Figure S21-5. Information Related to Channels in I/O Circuit Diagrams

S21.2.7 Update Circuit Diagrams for Consumers from Database

Performing the **Update from Database** command on a circuit diagram results in an update of designations, cross-references and additional technical information in the diagram according to the actual information in the database of the Electrical Diagram Builder.

Appendix A Designation of Advant Objects

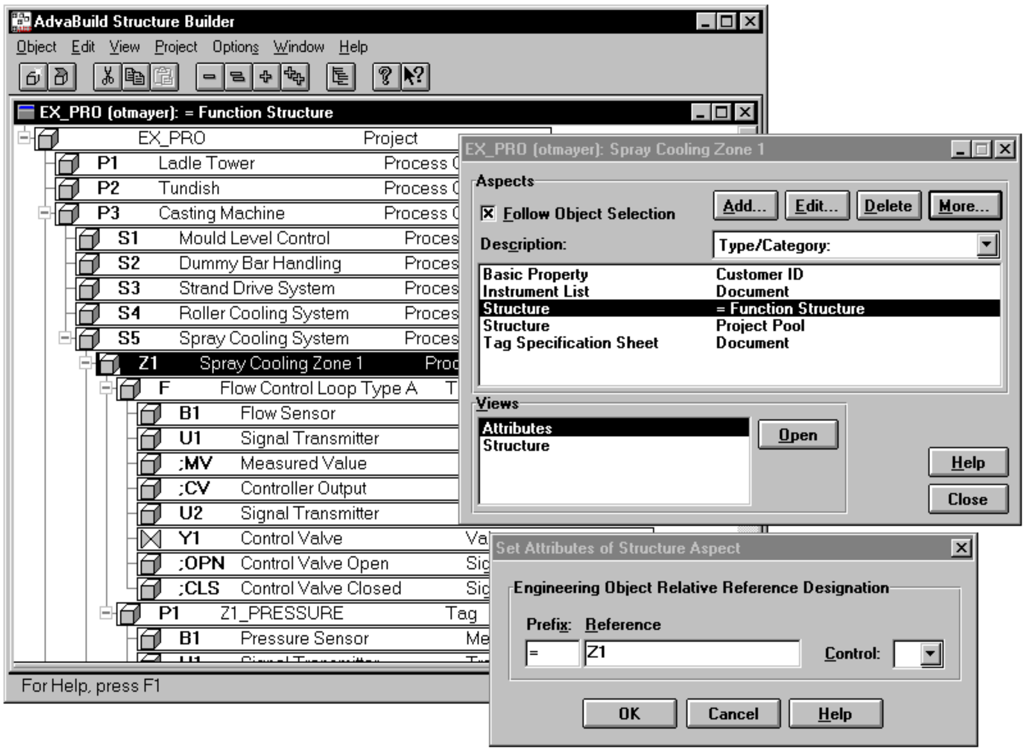
A.1 Applying the Reference Designation

Each Advant object should have a relative reference designation. Relative means that the Advant object is unambiguous designated with reference to its parent. A letter code according IEC 1346 should be used for the relative reference designation of products and components. The letter code is followed by a numerical part.

The concatenation of the relative reference designation within a structure path shows us the absolute reference designation. The absolute reference designation designates an Advant object unambiguously within the plant.

Inserting the Relative Reference Designation	
Preconditions:	Advant object created
User activities:	1. Use the Structure Builder 2. Open Advant Object 3. Select **Structure** aspect of the structure in which you work 4. Open **Attribute** view 5. Edit **Reference Designation** and **Prefix** in the **Set Attributes** window 6. Click on **OK**
Result:	Relative Reference Designation assigned to the Advant Object

Figure A-1 shows an open Advant object with its object window and the open Set Attributes window.


AdvaBuild Structure Builder
Object Edit View Project Options Window Help
EX_PRO (otmayer): = Function Structure
For Help, press F1
EX_PRO (otmayer): Spray Cooling Zone 1
Aspects
Add... Edit... Delete More...
Views
Open
Help
Close
Set Attributes of Structure Aspect
OK Cancel Help

EX_PRO Project
P1 Ladle Tower Process (
P2 Tundish Process (
P3 Casting Machine Process (
S1 Mould Level Control Proces
S2 Dummy Bar Handling Proces
S3 Strand Drive System Proces
S4 Roller Cooling System Proces
S5 Spray Cooling System Proces
Z1 Spray Cooling Zone 1 Proc
F Flow Control Loop Type A T
B1 Flow Sensor
U1 Signal Transmitter
;MV Measured Value
;CV Controller Output
U2 Signal Transmitter
Y1 Control Valve Va
;OPN Control Valve Open Sig
;CLS Control Valve Closed Sig
P1 Z1_PRESSURE Tag
B1 Pressure Sensor Me

☒ Follow Object Selection

Description:	Type/Category:
Basic Property	Customer ID
Instrument List	Document
Structure	= Function Structure
Structure	Project Pool
Tag Specification Sheet	Document

Views: Attributes, Structure

Engineering Object Relative Reference Designation

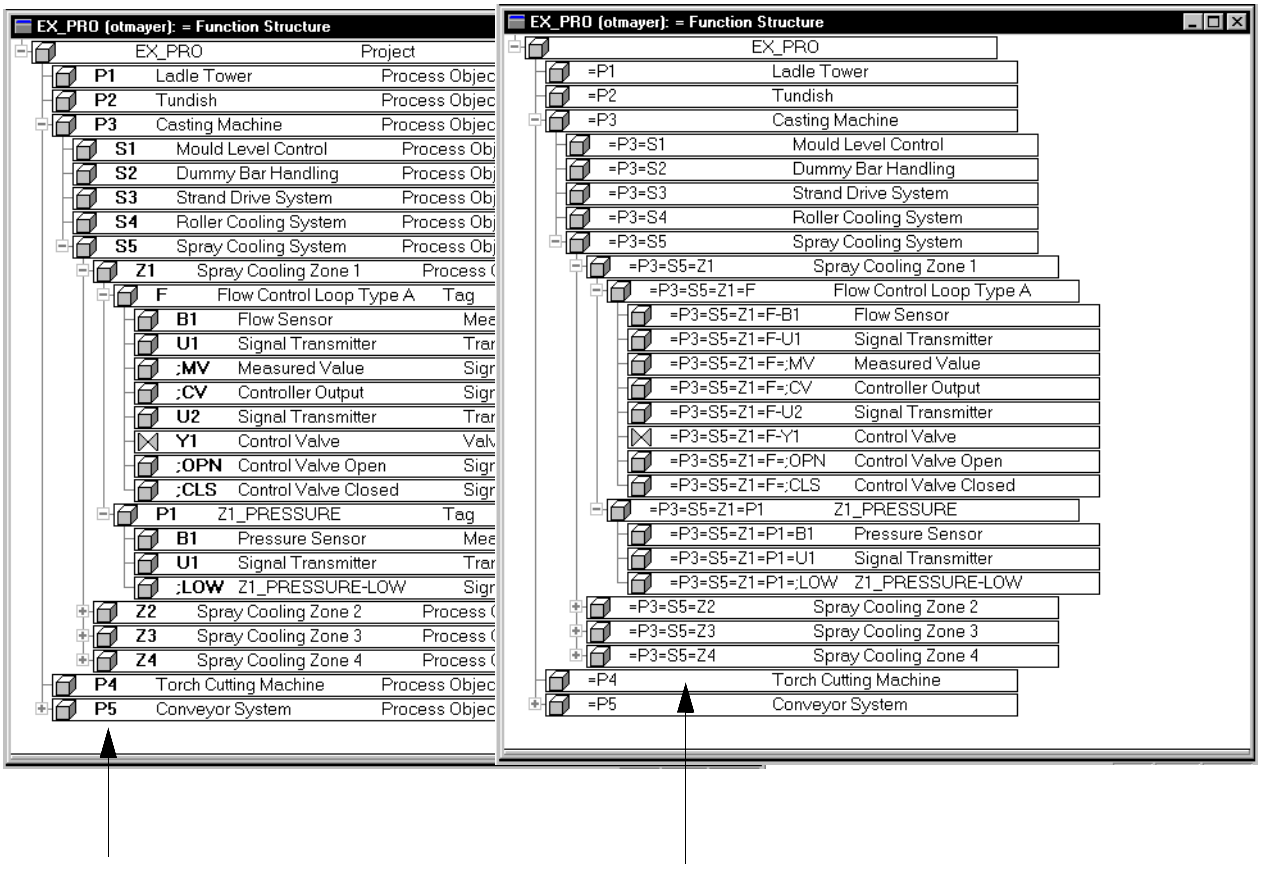
Prefix: =
Reference: Z1
Control:

Figure A-1. Editing the Relative Reference Designation

A.2 Presentation of the Absolute Reference Designation

The absolute reference designation is created automatically. It is an attribute of the aspect type structure and can be presented on the Advant object display.

Figure A-2 shows a Function Oriented Structure with display of the relative reference designation and the same structure with the absolute reference designation displayed.



EX_PRO (otmayer): = Function Structure

	EX_PRO	Project
P1	Ladle Tower	Process Objec
P2	Tundish	Process Objec
P3	Casting Machine	Process Objec
S1	Mould Level Control	Process Obj
S2	Dummy Bar Handling	Process Obj
S3	Strand Drive System	Process Obj
S4	Roller Cooling System	Process Obj
S5	Spray Cooling System	Process Obj
Z1	Spray Cooling Zone 1	Process
F	Flow Control Loop Type A	Tag
B1	Flow Sensor	Mea
U1	Signal Transmitter	Trar
;MV	Measured Value	Sigr
;CV	Controller Output	Sigr
U2	Signal Transmitter	Trar
Y1	Control Valve	Valv
;OPN	Control Valve Open	Sigr
;CLS	Control Valve Closed	Sigr
P1	Z1_PRESSURE	Tag
B1	Pressure Sensor	Mea
U1	Signal Transmitter	Trar
;LOW	Z1_PRESSURE-LOW	Sigr
Z2	Spray Cooling Zone 2	Process
Z3	Spray Cooling Zone 3	Process
Z4	Spray Cooling Zone 4	Process
P4	Torch Cutting Machine	Process Objec
P5	Conveyor System	Process Objec

EX_PRO (otmayer): = Function Structure

EX_PRO	
=P1	Ladle Tower
=P2	Tundish
=P3	Casting Machine
=P3=S1	Mould Level Control
=P3=S2	Dummy Bar Handling
=P3=S3	Strand Drive System
=P3=S4	Roller Cooling System
=P3=S5	Spray Cooling System
=P3=S5=Z1	Spray Cooling Zone 1
=P3=S5=Z1=F	Flow Control Loop Type A
=P3=S5=Z1=F-B1	Flow Sensor
=P3=S5=Z1=F-U1	Signal Transmitter
=P3=S5=Z1=F=;MV	Measured Value
=P3=S5=Z1=F=;CV	Controller Output
=P3=S5=Z1=F-U2	Signal Transmitter
=P3=S5=Z1=F-Y1	Control Valve
=P3=S5=Z1=F=;OPN	Control Valve Open
=P3=S5=Z1=F=;CLS	Control Valve Closed
=P3=S5=Z1=P1	Z1_PRESSURE
=P3=S5=Z1=P1=B1	Pressure Sensor
=P3=S5=Z1=P1=U1	Signal Transmitter
=P3=S5=Z1=P1=;LOW	Z1_PRESSURE-LOW
=P3=S5=Z2	Spray Cooling Zone 2
=P3=S5=Z3	Spray Cooling Zone 3
=P3=S5=Z4	Spray Cooling Zone 4
=P4	Torch Cutting Machine
=P5	Conveyor System

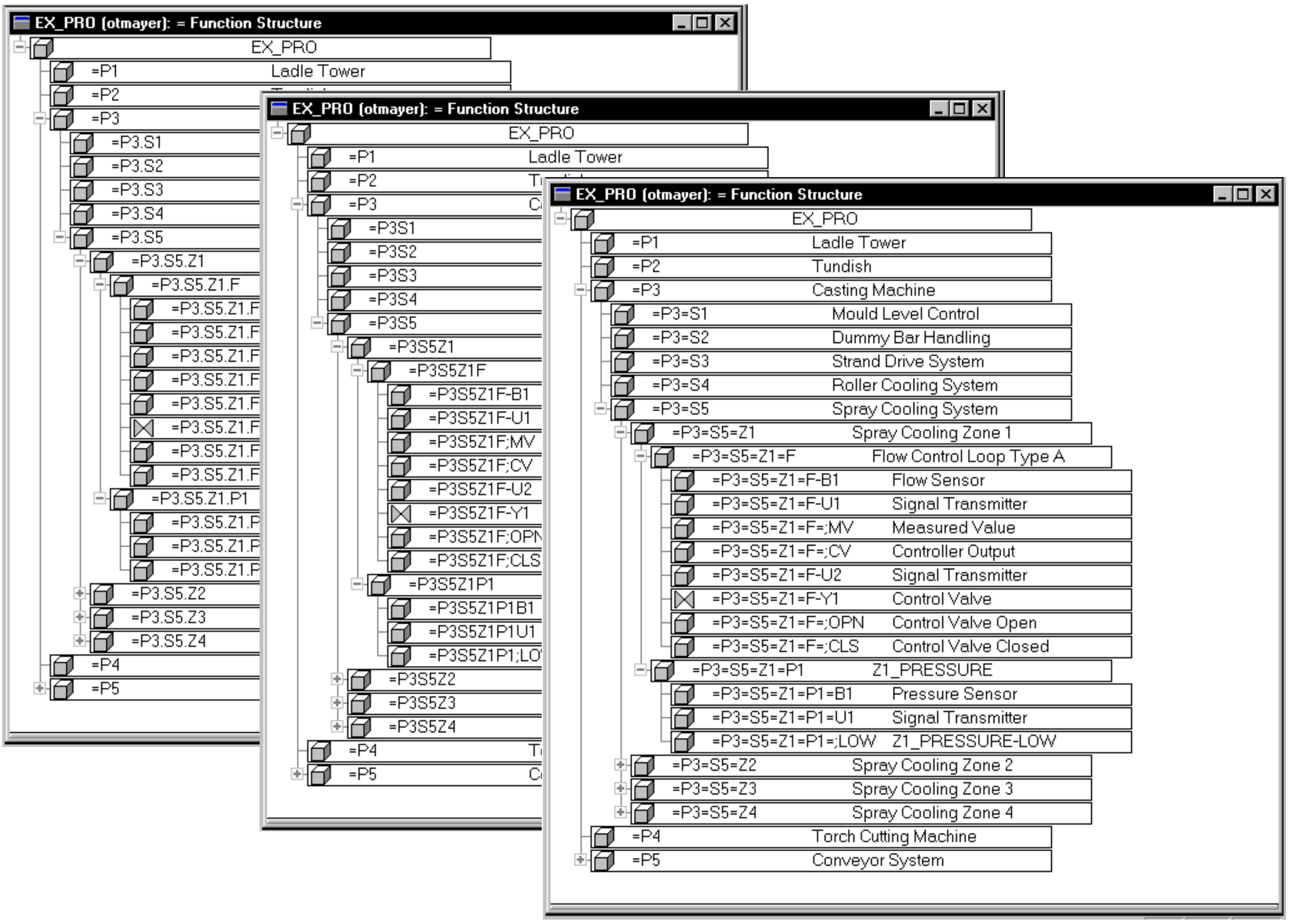
Relative Reference Designation

Absolute Reference Designation

Figure A-2. Relative and Absolute Reference Designation

Three types of presentation are selectable for the absolute reference designation. They differ in use of separators between the concatenated relative reference designations. The choice is between the use of the prefix of the relative ref. designation as separator or of a dot as separator or to use no separators.

Figure A-3 shows the available presentation of the absolute reference designations.



EX_PRO (otmayer): = Function Structure

- EX_PRO
  - =P1 Ladle Tower
  - =P2
  - =P3
    - =P3.S1
    - =P3.S2
    - =P3.S3
    - =P3.S4
    - =P3.S5
      - =P3.S5.Z1
        - =P3.S5.Z1.F
          - =P3.S5.Z1.F
          - =P3.S5.Z1.F
          - =P3.S5.Z1.F
          - =P3.S5.Z1.F
          - =P3.S5.Z1.F
          - =P3.S5.Z1.F
          - =P3.S5.Z1.F
          - =P3.S5.Z1.F
        - =P3.S5.Z1.P1
          - =P3.S5.Z1.P
          - =P3.S5.Z1.P
          - =P3.S5.Z1.P
      - =P3.S5.Z2
      - =P3.S5.Z3
      - =P3.S5.Z4
  - =P4
  - =P5

EX_PRO (otmayer): = Function Structure

- EX_PRO
  - =P1 Ladle Tower
  - =P2 T
  - =P3 C
    - =P3S1
    - =P3S2
    - =P3S3
    - =P3S4
    - =P3S5
      - =P3S5Z1
        - =P3S5Z1F
          - =P3S5Z1F-B1
          - =P3S5Z1F-U1
          - =P3S5Z1F;MV
          - =P3S5Z1F;CV
          - =P3S5Z1F-U2
          - =P3S5Z1F-Y1
          - =P3S5Z1F;OPN
          - =P3S5Z1F;CLS
        - =P3S5Z1P1
          - =P3S5Z1P1B1
          - =P3S5Z1P1U1
          - =P3S5Z1P1;LO
      - =P3S5Z2
      - =P3S5Z3
      - =P3S5Z4
  - =P4 T
  - =P5 C

EX_PRO (otmayer): = Function Structure

- EX_PRO
  - =P1 Ladle Tower
  - =P2 Tundish
  - =P3 Casting Machine
    - =P3=S1 Mould Level Control
    - =P3=S2 Dummy Bar Handling
    - =P3=S3 Strand Drive System
    - =P3=S4 Roller Cooling System
    - =P3=S5 Spray Cooling System
      - =P3=S5=Z1 Spray Cooling Zone 1
        - =P3=S5=Z1=F Flow Control Loop Type A
          - =P3=S5=Z1=F-B1 Flow Sensor
          - =P3=S5=Z1=F-U1 Signal Transmitter
          - =P3=S5=Z1=F=;MV Measured Value
          - =P3=S5=Z1=F=;CV Controller Output
          - =P3=S5=Z1=F-U2 Signal Transmitter
          - =P3=S5=Z1=F-Y1 Control Valve
          - =P3=S5=Z1=F=;OPN Control Valve Open
          - =P3=S5=Z1=F=;CLS Control Valve Closed
        - =P3=S5=Z1=P1 Z1_PRESSURE
          - =P3=S5=Z1=P1=B1 Pressure Sensor
          - =P3=S5=Z1=P1=U1 Signal Transmitter
          - =P3=S5=Z1=P1=;LOW Z1_PRESSURE-LOW
      - =P3=S5=Z2 Spray Cooling Zone 2
      - =P3=S5=Z3 Spray Cooling Zone 3
      - =P3=S5=Z4 Spray Cooling Zone 4
  - =P4 Torch Cutting Machine
  - =P5 Conveyor System

Figure A-3. Display Variants for the Absolute Reference Designation

3BSE 013 933R201

September 1999